The Mythical Pegasus and Gravitar

Deep Learning and Reinforcement Learning

Introduction

The assignment is split into two parts: (1) Deep Learning (50 marks) and (2) Reinforcement Learning (50 marks) accordingly, for a combined total of 100 marks available.

In short, you are to build a generative model that creates unique and diverse images of winged horses that all look like a Pegasus, the mythical divine horse in Greek mythology. This is usually depicted as pure white, with clear outspread wings. The second part of the assignment is to train a deep reinforcement learning agent to play the Atari 2600 game Gravitar, published in 1983. This is one of the most difficult Atari games where you have to fly to different planets as you get pulled towards deadly stars by gravity, shoot enemies, conserve and pick up fuel with a tractor beam that also deflects bullets.

For the Pegasus part, you are to write a short scientific paper for the method, results, and limitations in a provided LaTeX template that closely follows parts of the ICLR 2021 conference style guidelines. Whereas for the Gravitar part, you are to provide a video of the agent playing the game, code, and log file (no report is required). These files must all be zipped together like this:

```
submission.zip
   pegasus-paper.pdf
   pegasus-code.ipynb (or .py)
   gravitar-code.ipynb (or .py)
   gravitar-video-episode-1210-score-85.mp4
   gravitar-log.txt
```

To assist in this, the following templates are provided to build on:

- [Pegasus Paper LaTeX Template]
- [Google Colab Pegasus Starter Code]
- [Google Colab Gravitar Starter Code]

Part 1: The Mythical Pegasus

Using the CIFAR-10 and/or STL-10 datasets, you are to train a deep generative model to synthesise a unique batch of 64 images that all looked like winged horses (of any colour). There are no such images of winged horses in these two datasets. You should then select, from the batch, the single best image that looks most like a white Pegasus. The paper should use the provided LaTeX template, where you are to write up the methodology, results, and limitations of your approach alongside a short abstract.

The report should be written like an academic paper, where mathematical notation should try to follow the ICLR 2021 guidelines (see the template for more information). This means your discussions should be short, clear, and concise —*less is more*. Where appropriate, it is recommended to include a high-level architectural diagram in the paper to help explain your approach.

You can use any architecture that you like, and you can use any sampling strategy. For example you could train an autoencoder, and condition it on a birds and horses which have been manually identified from the dataset. These datasets mostly have annotated class labels available for the images, which you can use to help identify the horses and birds. This will give two latent codes via the encoder network, that you could linearly interpolate between to give the final outputs via the decoder network. Alternatively, you

could train a GAN and randomly sample a batch of 64 images from the model. These two ideas are not necessarily the best solution. Also, there are penalties and bonuses that will influence your design:

Use any adversarial training (e.g. GAN) method	-4 marks
Your best Pegasus is non-white (e.g. brown or black)	-2 marks
Nearly all winged horses in the batch are white	$+1$ mark
Train only on CIFAR-10	-2 marks
Train with STL-10 resized to 48x48 pixels	$+1$ mark
Train with STL-10 resized to 64x64 pixels	$+1$ mark
Train with STL-10 at the full 96x96 pixels	$+3$ marks
Manually edit (paint) any images or outputs	-50 marks
Train on any other data outside of the datasets	-50 marks
Use or modify someones code without referencing it	-50 marks
Use pre-trained weights from another model	-50 marks
Every page over 4 pages in the paper (excluding references)	-5 marks

Table 1: Penalties and bonuses accumulate, and are added onto the final mark.

Please state at the end of the paper the total bonus or penalty you are expecting, based on this table. For example if you successfully train a GAN to produce a white Pegasus from a batch of mixed-colour winged horses, using both data from CIFAR-10 and from STL-10 at the full resolution, you can expect to receive: -4 marks (as its a GAN), then 1+1+3 = +5 marks for STL-10 at 96x96 resolution, for a total bonus of +1 mark. If you submit a paper that is 7 pages long, you will receive an additional -15 mark penalty.

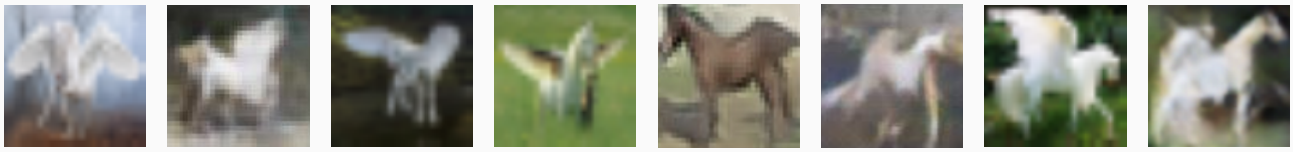


Figure 1: Generating a realistic and recognisable Pegasus is very difficult within the dataset constraints, therefore outstanding attempts with unimpressive results can still get good marks. These are some excellent results when training only on CIFAR-10.

Pegasus paper marking scheme

The paper will be marked as follows:

- **[20 marks]** Scientific quality and mathematical rigor for the paper and solution
  - Communication, application, and presentation of the underpinning mathematical theory
  - Architectural design, sophistication, appropriateness, and novelty
  - Clarity, simplicity and frugality of both the scientific writing and the implementation
- **[10 marks]** Recognisability of the single best output
  - Can I tell this is an image of a Pegasus?
  - How much do I need to stretch my imagination to see a winged horse?
- **[10 marks]** Realism of the sampled batch of model outputs
  - Are the generated images blurry?
  - Do the objects in them have realistic shapes or textures? Do they look real?
- **[10 marks]** Uniqueness of the sampled batch of model outputs
  - How different are the images from their nearest neighbours in the dataset?
  - How diverse are the samples within the batch of 64 provided?
  - Do all the winged horses look the same? Is there any mode collapse?

Part 2: Gravitar

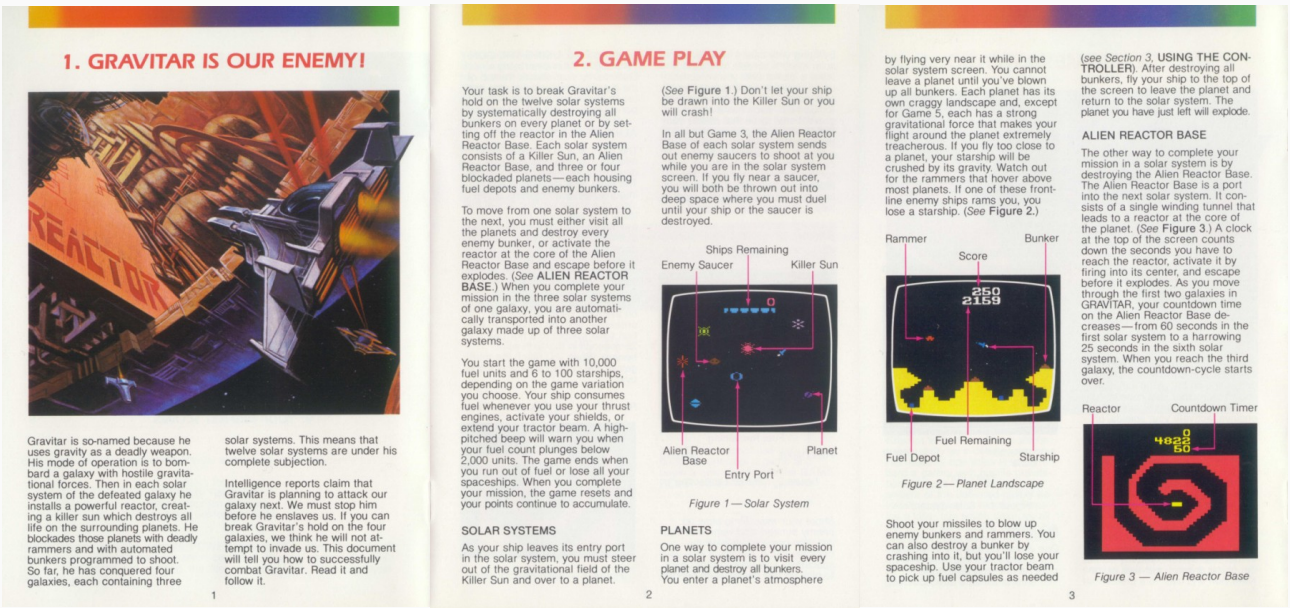
Gravitar is a notoriously difficult Atari 2600 game, with complex controls and changing environments where you always have to navigate away from gravity. This is what the beautiful Atari 2600 console looked like, which was released in 1977. It had an 8-bit 1.19 MHz CPU with just 128 bytes of RAM!

Figure 2: Atari 2600 console

Figure 3: Gravitar cartridge

I recommend that before you begin, you have a go at playing Gravitar online to understand the game mechanics. Try to visit a couple of planets and use the tractor beam (down key) to pick up some more fuel. My score is 7,250 - can you do any better?

- [Play Gravitar with an online Atari 2600 emulator] spacebar to start/fire and move with arrow keys
- [More information, including the full manual and cartridge images]



1. GRAVITAR IS OUR ENEMY!

Gravitar is so-named because he uses gravity as a deadly weapon. His mode of operation is to bombard a galaxy with hostile gravitational forces. Then in each solar system of the defeated galaxy he installs a powerful reactor, creating a killer sun which destroys all life on the surrounding planets. He blockades those planets with deadly rammers and with automated bunkers programmed to shoot. So far, he has conquered four galaxies, each containing three solar systems. This means that twelve solar systems are under his complete subjection.

Intelligence reports claim that Gravitar is planning to attack our galaxy next. We must stop him before he enslaves us. If you can break Gravitar's hold on the four galaxies, we think he will not attempt to invade us. This document will tell you how to successfully combat Gravitar. Read it and follow it.

1

2. GAME PLAY

Your task is to break Gravitar's hold on the twelve solar systems by systematically destroying all bunkers on every planet or by setting off the reactor in the Alien Reactor Base. Each solar system consists of a Killer Sun, an Alien Reactor Base, and three or four blockaded planets—each housing fuel depots and enemy bunkers.

To move from one solar system to the next, you must either visit all the planets and destroy every enemy bunker, or activate the reactor at the core of the Alien Reactor Base and escape before it explodes. (See ALIEN REACTOR BASE.) When you complete your mission in the three solar systems of one galaxy, you are automatically transported into another galaxy made up of three solar systems.

You start the game with 10,000 fuel units and 6 to 100 starships, depending on the game variation you choose. Your ship consumes fuel whenever you use your thrust engines, activate your shields, or extend your tractor beam. A high-pitched beep will warn you when your fuel count plunges below 2,000 units. The game ends when you run out of fuel or lose all your spaceships. When you complete your mission, the game resets and your points continue to accumulate.

SOLAR SYSTEMS

As your ship leaves its entry port in the solar system, you must steer out of the gravitational field of the Killer Sun and over to a planet. (See Figure 1.) Don't let your ship be drawn into the Killer Sun or you will crash!

In all but Game 3, the Alien Reactor Base of each solar system sends out enemy saucers to shoot at you while you are in the solar system screen. If you fly near a saucer, you will both be thrown out into deep space where you must duel until your ship or the saucer is destroyed.

Ships Remaining
Enemy Saucer
Killer Sun
Alien Reactor Base
Planet
Entry Port

Figure 1—Solar System

PLANETS

One way to complete your mission in a solar system is to visit every planet and destroy all bunkers. You enter a planet's atmosphere by flying very near it while in the solar system screen. You cannot leave a planet until you've blown up all bunkers. Each planet has its own craggy landscape and, except for Game 5, each has a strong gravitational force that makes your flight around the planet extremely treacherous. If you fly too close to a planet, your starship will be crushed by its gravity. Watch out for the rammers that hover above most planets. If one of these frontline enemy ships rams you, you lose a starship. (See Figure 2.)

2

Rammer
Bunker
Score
250
2159
Fuel Remaining
Fuel Depot
Starship

Figure 2—Planet Landscape

Shoot your missiles to blow up enemy bunkers and rammers. You can also destroy a bunker by crashing into it, but you'll lose your spaceship. Use your tractor beam to pick up fuel capsules as needed (see Section 3, USING THE CONTROLLER). After destroying all bunkers, fly your ship to the top of the screen to leave the planet and return to the solar system. The planet you have just left will explode.

ALIEN REACTOR BASE

The other way to complete your mission in a solar system is by destroying the Alien Reactor Base. The Alien Reactor Base is a port into the next solar system. It consists of a single winding tunnel that leads to a reactor at the core of the planet. (See Figure 3.) A clock at the top of the screen counts down the seconds you have to reach the reactor, activate it by firing into its center, and escape before it explodes. As you move through the first two galaxies in GRAVITAR, your countdown time on the Alien Reactor Base decreases—from 60 seconds in the first solar system to a harrowing 25 seconds in the sixth solar system. When you reach the third galaxy, the countdown-cycle starts over.

Reactor
Countdown Timer
4822
50

Figure 3 — Alien Reactor Base

3

Task

Your task is to build and train a deep reinforcement learning agent to obtain the highest score possible, using the provided starter code linked on the first page. OpenAI gym provides two Gravitar environments '**Gravitar-ram-v0**' and '**Gravitar-v0**' respectively. You can use either of them, but pay attention to the size of the input array shapes as documented here and here.

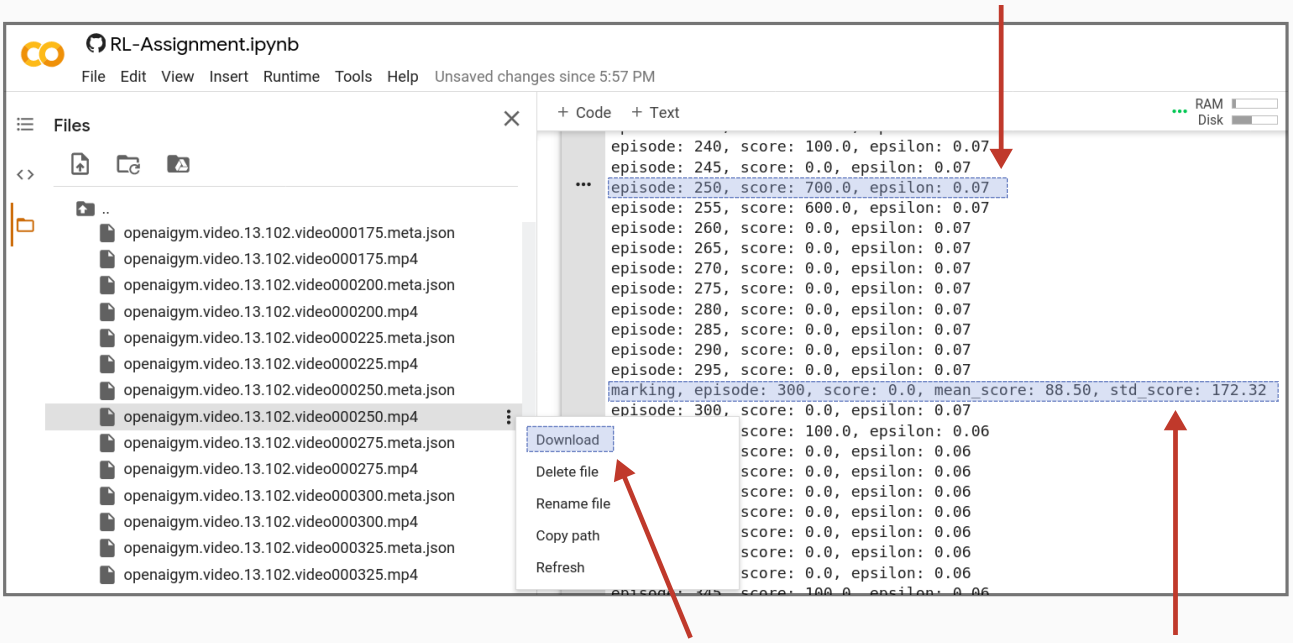
You can use any reinforcement learning algorithm that you like, including ones not covered in the lectures.

Submission

The following figure explains how and what to submit. You can print whatever data you like into the log, but you must keep the string which starts with 'marking'. This string prints the mean and standard deviation of the score over every 100 episodes. You must not change this. The submitted video should show the highest score that you are able to successfully record. Recording a video is quite slow, so you may wish

to do it every 25 or more episodes. Increasing this interval may slightly improve training performance, but you may miss recording a good episode.

This is the best episode with a recorded video



Therefore download episode 250 and rename it **gravitar-video-episode-250-score-700.mp4**

Make sure these lines are printed
Copy the full log to **gravitar-log.txt**

The code submission must be written with clarity and minimalism. You can submit an .ipynb or .py file. You do not need to follow PEP-8 or any departmental guidelines for code quality with this assignment. Try to keep comments to a minimum, as good code should speak for itself. If you have tried many impressive designs and ideas, you may briefly explain these in a large comment (or text in .ipynb if you prefer) at the bottom of the submission, but do not expect this to significantly influence marking of your final solution.

Anti-tampering

There will be statistical analysis of the submitted log files. Any log files with outliers in the convergence behaviour will have their associated code retrained multiple times using SLURM scripts on NCC, learning a distribution over its expected convergence behaviour. If the likelihood of the submitted log data shows evidence of tampering, a departmental investigation will be conducted. If interested, this method for detecting tampering is similar to parts of the solution used to solve Sploosh Kaboom in Zelda Windwaker.

Marking

The code, video, and log submission will be marked as follows:

- **[30 marks]** Convergence and score
  - How efficiently does your agent learn?
  - Does it just get lucky after lots and lots of training?
  - How often does it get a good score?
- **[10 marks]** Sophistication and appropriateness of the solution
  - How well have you applied the relevant theory to the problem?
  - How hackish is your implementation, or is it robust and well-designed?
  - Have you just cited and pasted code, or is their evidence of comprehension with further study and novel design extending beyond the lecture materials?
- **[10 marks]** Intuition of video
  - Does the video look like the agent is just randomly getting lucky after lots of episodes?

- Is it chasing and hunting down the enemies efficiently?
- Is it navigating gravity gracefully, or is it like it's powered by an infinite improbability drive?
- Has it learnt any surprising behaviours?

PyTorch Training

This assignment can be completed entirely using Google Colab, or you may wish to register for an account and train on NCC. Users without remote server experience or job queuing system experience (such as SLURM) are recommended to continue to use Google Colab, which is just as fast for PyTorch training. If using NCC, please carefully read the documentation and respect other users on the job queuing system.

Closing Comment

I hope that you enjoy this coursework. If you are struggling, please join one of the weekly zoom meetings to ask questions and discuss any issues, such as with programming or relevant theory.

Questions and Answers

It is strongly recommended to read these common questions and answers carefully.

I've spent a million hours training these models and I still need more time. This assignment is impossible!	This is always the problem with Deep Learning and Reinforcement Learning. Some of our models take many months to train, so you need to choose your experiments very carefully, with only well-tested code where you have evidence it will converge given more time. Also, in some ways, yes these tasks are impossible to get to perfection. That is the nature of these ill-defined learning problems, but i'm not looking for perfection for full marks. When marking, I will be looking carefully at the expected converging behaviour of the method, rather than directly at the best scores/images.
I just read this interesting new paper which is very different to techniques from the course. Can I use it?	Yes! This is exactly the kind of thing you should be doing. Study the paper and reimplement it, but make sure you cite it in the report, and you must also reference it in the comments at the top of the code.
I found code online which looks similar to what I need. Can I use it?	Yes, but you must cite the code in both the written report and in the comments at the top of the code. I will then carefully cross-reference this code with your implementation to see how well you have adapted their code. If you have simply copied and pasted without evidence of experimentation or tailoring, do not expect to get a good grade even if your results are good. Whereas if I see evidence of original interpretation, novel comprehension and application of the theory in the lectures, even if the experimental results are not as strong, you can get very high marks. However if you pass-off other people's code as your own, and 'forget' to cite their code, or work together with other students, you will very likely get caught (see the submission Plagiarism and Collusion section on DUO to read about the tools used to detect this). This incurs a very severe departmental penalty.

Table 2: Common questions and answers

“	What do you mean by manual image editing? Can I write code to do X?
💬	The answer to this is ‘within reason’. If you manually paint wings on the inputs or outputs, you will get zero. If you write code to manually set the pixels which look like wings, you’ll also get zero. If you start using the OpenCV library to enhance and brighten the images etc, then technically yes this is acceptable, but do not expect to get additional marks for this, as this is a Deep Learning assignment and not Computer Vision or Image Processing.
“	Isn’t the best strategy to just copy the state-of-the-art?
💬	Actually no, the state-of-the-art for these tasks are large distributed models (which use large numbers of GPUs scaled horizontally). You don’t have these kinds of resources available to you, so you will need a different strategy. The state-of-the-art models also have been highly tuned to squeeze every last bit of juice out of them. If you copy this code directly, as mentioned in a previous answer, you won’t get a good grade. Also I will be marking based on the convergence behaviour, not just the final score. If I find evidence that a poor quality model has been trained with large numbers of GPUs for several thousand pounds on AWS, it will not be awarded higher marks than an excellent model that has only been trained for a few hours on a single Google Colab GPU – even if it has a higher score.
“	To farm the STL-10 bonuses, can I train on CIFAR-10, then do one optimisation step at the full resolution on STL-10, so I can say i’ve trained on it?
💬	This won’t make significant improvement to the output quality, so I won’t count the bonuses for this.
“	Do I have to train on all the data?
💬	No, you can train on parts of CIFAR-10 or STL-10, or combinations of them in any way you like.
“	I have this amazing/funny/great video. The score is not as good, but it’s so surprising! Can I submit it too?
💬	Yes, you can upload an additional outtake video named accordingly to this: **gravitar-outtake-episode-100-score-50.mp4**
“	What are pre-trained weights?
💬	This is where you would use a model that someone else has trained on different data, e.g. a GAN trained on ImageNet, and then transfer the weights (transfer learning) for the new task. While this would generate beautiful high-resolution winged horses, this is not allowed.
“	Can I use Keras, Theano, Tensorflow, Caffe, Chainer, or X to develop the coursework instead?
💬	No, unless you already are very experienced with PyTorch and want to use this assignment to learn another framework such as JAX. If this is the case, you must email me and provide evidence that you already have significant PyTorch experience, as it is the industry standard for DL research.

Table 2: Common questions and answers

“	I’m really struggling and feeling overwhelmed by all of this. The maths is too complicated and I don’t know where to begin.
💬	Okay, first don’t panic, get a wet towel and wipe your face :) Now open up clean versions of the Colab Starter Code and try running it. Next try to make some small modifications to the models. If you get errors, read them slowly, Google them. Start by trying to improve the existing code – why not try making it residual to start with? Try to get a feel for its limitations. When you’re confident enough, try to implement something a little bit more sophisticated that you think will do better. Here you need to be careful, don’t try too much at once. Try to add just the smallest bit of extra functionality you can, then test it and check if it works. Build up slowly and incrementally until you have the confidence to try more advanced ideas.
“	Is Google Colab free to use?
💬	Yes, although when Google is experiencing heavy load they impose limits (which vary) on their resources. Therefore to get the most out of Colab, save your model and optimisation parameters as in the example code, and close your tabs when you are done. This is because Google prioritises users who have used less resources.
“	What Gravitar score do I need for a 1st? Will this Pegasus give me a 1st?
💬	I can’t answer this, as the marking ‘function’ has too many parameters.
“	Does the page limit include references?
💬	No, your references section can start on the 5th page without penalty. You can have an unlimited number of references.
“	Do I have to use the LaTeX template for the paper?
💬	Yes, this is for the experience of writing a conference paper in this field and it will hopefully encourage slower and more concise writing, alongside mathematical rigor with professional typesetting.
“	Help! I’ve never used or been taught LaTeX!
💬	After you’ve logged into overleaf.com and cloned the paper template (click the copy icon from the overleaf.com home page, then edit the copied version), just modify some of the text and press Ctrl+S to compile. If you want to achieve something specific, Google it. You can then download and submit the compiled .PDF directly from overleaf. The rest will come naturally.

Table 2: Common questions and answers

介绍

作业分为两部分：（1）深度学习（50 分）和（2）强化学习（50 分），总共有 100 分。
简而言之，您将要建立一个生成模型，以创建独特而多样的有翼马图像，它们看起来都像希腊神话中的神话般的神马飞马。通常将其描绘为纯白色，翅膀清晰展开。任务的第二部分是训练一名深度强化学习代理，以玩 Atari 2600 游戏 Gravitar，该游戏于 1983 年发布。这是最困难的 Atari 游戏之一，当您被引向致命的恒星时，您必须飞往不同的星球在重力作用下射击敌人，保存并用能使子弹偏转的牵引机光束收集燃料。
对于 Pegasus 部分，您将在提供的 LATEX 模板中针对方法，结果和局限性撰写一篇简短的科学论文，该模板密切遵循 ICLR 2021 会议风格指南的各个部分。而对于 Gravitar 部分，您将提供正在玩游戏的座席的视频，代码和日志文件（不需要报告）。这些文件必须全部压缩在一起，如下所示：
Submit.zip
pegasus-paper.pdf
pegasus-code.ipynb（或.py）
gravitar-code.ipynb（或.py）
gravitar-video-episode-1210-score-85.mp4
gravitar-log.txt

为了帮助完成此任务，提供了以下模板作为基础：
W [Pegasus Paper LATEX Template] -见附件
W [Google Colab Pegasus 入门代码]- DL-Assignment.ipynb - Colaboratory (google.com)
W [Google Colab Gravitar 入门代码]- RL-Assignment.ipynb - Colaboratory (google.com)

第 1 部分：神话中的飞马座

使用 CIFAR-10 和/或 STL-10 数据集，您将训练一个深度生成模型，以合成一批独特的 64 张图像，这些图像看上去都像有翼马（任何颜色）。在这两个数据集中，没有此类有翼马的图像。然后，您应该从批次中选择最像白色飞马的最佳图像。本文应使用提供的 LATEX 模板，在其中编写方法论，结果以及方法的局限性以及简短的摘要。
报告的撰写应像学术论文一样，其中数学符号应尝试遵循 ICLR 2021 准则（有关更多信息，请参见模板）。这意味着您的讨论应该简短，清晰，简洁-少即是多。建议在适当的地方在本文中包含一个高级体系结构图，以帮助解释您的方法。
您可以使用任何喜欢的架构，也可以使用任何采样策略。例如，您可以训练自动编码器，并将其设置为从数据集中手动识别的鸟和马。这些数据集大多具有可用于图像的带注释的类标签，您可以使用它们来识别马和鸟。这将通过编码器网络提供两个潜在代码，您可以在它们之间进行线性内插，以通过解码器网络提供最终输出。或者，您可以训练 GAN 并从模型中随机采样一批 64 张图像。这两个想法不一定是最佳解决方案。此外，还有一些罚款和奖金会影响您的设计：

使用任何对抗训练（例如 GAN）方法（−4 分）
最好的飞马座是非白色的（例如棕色或黑色）（-2 分）
批次中几乎所有有翼马都是白色的（+1 分）
仅在 CIFAR-10 上训练（−2 分）

使用 STL-10 调整大小为 48x48 像素的训练（+1 标记）
使用 STL-10 调整大小为 64x64 像素的训练（+1 标记）
使用 STL-10 以全 96x96 像素（+3 标记）进行训练
手动编辑（绘制）任何图像或输出（−50 标记）
训练数据集以外的任何其他数据（−50 分）
使用或修改某人的代码而不参考它（−50 标记）
使用其他模型的预训练权重（−50 标记）
论文中每 4 页以上的每一页（不包括参考文献）（-5 分）
表 1
表 1：罚款和奖金累积，并加到最终分数上。

请根据此表在文件末尾注明您期望的总奖金或罚款。 例如，如果您成功地使用一组来自 CIFAR-10 和 STL-10 的数据以全分辨率成功地训练了 GAN 从一批混色有翼马匹中生产出白色飞马，您可以期望获得：-4 分（ 作为其 GAN），那么对于 STL-10，在 96x96 分辨率下为 1 1 3 = 5 分，总奖金为 1 分。 如果您提交的论文长达 7 页，您将受到-15 分的罚款。

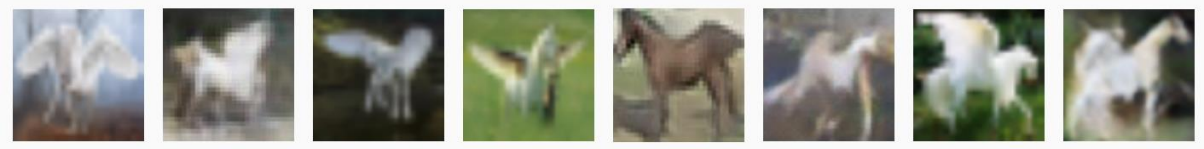


图 1
图 1：在数据集约束内生成逼真且可识别的 Pegasus 非常困难，因此，效果令人印象深刻的出色尝试仍会获得良好的成绩。 当仅在 CIFAR-10 上进行训练时，这些结果是出色的。

飞马纸打分方案

该文件将被标记如下：

- [20 分]论文和解决方案的科学质量和数学严谨性
  - 基础数学理论的交流，应用和展示
  - 建筑设计，精致，适当和新颖
  - 科学写作和实施的清晰性，简洁性和节俭性
- [10 分]单个最佳输出的可识别性
  - 我可以说这是飞马的形象吗？
  - 要想看到一匹有翼的马，我需要加倍的想象力吗？
- [10 分]抽样模型输出的真实感
  - 生成的图像模糊吗？
  - 其中的对象是否具有逼真的形状或纹理？它们看起来真实吗？
- [10 分]抽样的模型输出批次的唯一性
  - 图像与数据集中最近的图像有何不同？
  - 提供的 64 个批次中的样本有多少种？
  - 所有的有翼马看起来都一样吗？有模式崩溃吗？

第 2 部分：Gravitar

Gravitar 是一款非常困难的 Atari 2600 游戏，具有复杂的控件和变化的环境，您必须始终远离重力。这就是 1977 年发布的漂亮的 Atari 2600 控制台的样子。它具有 8 位 1.19 MHz CPU 和仅 128 个字节的 RAM！

Figure 2: Atari 2600 console

Figure 3: Gravitar cartridge

图 2：Atari 2600 控制台图 3：Gravitar 墨盒

我建议您在开始之前先尝试在线玩 Gravitar，以了解游戏机制。 尝试拜访几个行星，然后使用牵引车横梁（向下键）拾取更多的燃料。 我的分数是 7,250-您能做得更好吗？

W [使用在线 Atari 2600 模拟器播放 Gravitar]空格键以启动/发射并使用箭头键移动-Javatari

W [更多信息，包括完整的手册和墨盒图像]- Atari 2600 VCS Gravitar : scans, dump, download, screenshots, ads, videos, catalog, instructions, roms (atarimania.com)

1. GRAVITAR IS OUR ENEMY!

Gravitar is so-named because he uses gravity as a deadly weapon. His mode of operation is to bombard a galaxy with hostile gravitational forces. Then in each solar system of the defeated galaxy he installs a powerful reactor, creating a killer sun which destroys all life on the surrounding planets. He blockades those planets with deadly rammers and with automated bunkers programmed to shoot. So far, he has conquered four galaxies, each containing three solar systems. This means that twelve solar systems are under his complete subjection.

Intelligence reports claim that Gravitar is planning to attack our galaxy next. We must stop him before he enslaves us. If you can break Gravitar's hold on the four galaxies, we think he will not attempt to invade us. This document will tell you how to successfully combat Gravitar. Read it and follow it.

1

2. GAME PLAY

Your task is to break Gravitar's hold on the twelve solar systems by systematically destroying all bunkers on every planet or by setting off the reactor in the Alien Reactor Base. Each solar system consists of a Killer Sun, an Alien Reactor Base, and three or four blockaded planets—each housing fuel depots and enemy bunkers.

To move from one solar system to the next, you must either visit all the planets and destroy every enemy bunker, or activate the reactor at the core of the Alien Reactor Base and escape before it explodes. (See **ALIEN REACTOR BASE**.) When you complete your mission in the three solar systems of one galaxy, you are automatically transported into another galaxy made up of three solar systems.

You start the game with 10,000 fuel units and 6 to 100 starships, depending on the game variation you choose. Your ship consumes fuel whenever you use your thrust engines, activate your shields, or extend your tractor beam. A high-pitched beep will warn you when your fuel count plunges below 2,000 units. The game ends when you run out of fuel or lose all your spaceships. When you complete your mission, the game resets and your points continue to accumulate.

SOLAR SYSTEMS

As your ship leaves its entry port in the solar system, you must steer out of the gravitational field of the Killer Sun and over to a planet. (See **Figure 1**.) Don't let your ship be drawn into the Killer Sun or you will crash!

In all but Game 3, the Alien Reactor Base of each solar system sends out enemy saucers to shoot at you while you are in the solar system screen. If you fly near a saucer, you will both be thrown out into deep space where you must duel until your ship or the saucer is destroyed.

Ships Remaining
Enemy Saucer
Killer Sun
Alien Reactor Base
Planet
Entry Port

Figure 1—Solar System

PLANETS

One way to complete your mission in a solar system is to visit every planet and destroy all bunkers. You enter a planet's atmosphere by flying very near it while in the solar system screen. You cannot leave a planet until you've blown up all bunkers. Each planet has its own craggy landscape and, except for Game 5, each has a strong gravitational force that makes your flight around the planet extremely treacherous. If you fly too close to a planet, your starship will be crushed by its gravity. Watch out for the rammers that hover above most planets. If one of these frontline enemy ships rams you, you lose a starship. (See **Figure 2**.)

Rammer
Bunker
Score
250
2159
Fuel Remaining
Fuel Depot
Starship

Figure 2—Planet Landscape

Shoot your missiles to blow up enemy bunkers and rammers. You can also destroy a bunker by crashing into it, but you'll lose your spaceship. Use your tractor beam to pick up fuel capsules as needed (see *Section 3*, **USING THE CONTROLLER**). After destroying all bunkers, fly your ship to the top of the screen to leave the planet and return to the solar system. The planet you have just left will explode.

ALIEN REACTOR BASE

The other way to complete your mission in a solar system is by destroying the Alien Reactor Base. The Alien Reactor Base is a port into the next solar system. It consists of a single winding tunnel that leads to a reactor at the core of the planet. (See **Figure 3**.) A clock at the top of the screen counts down the seconds you have to reach the reactor, activate it by firing into its center, and escape before it explodes. As you move through the first two galaxies in GRAVITAR, your countdown time on the Alien Reactor Base decreases—from 60 seconds in the first solar system to a harrowing 25 seconds in the sixth solar system. When you reach the third galaxy, the countdown-cycle starts over.

Reactor
Countdown Timer
40
00
150

Figure 3 — Alien Reactor Base

3

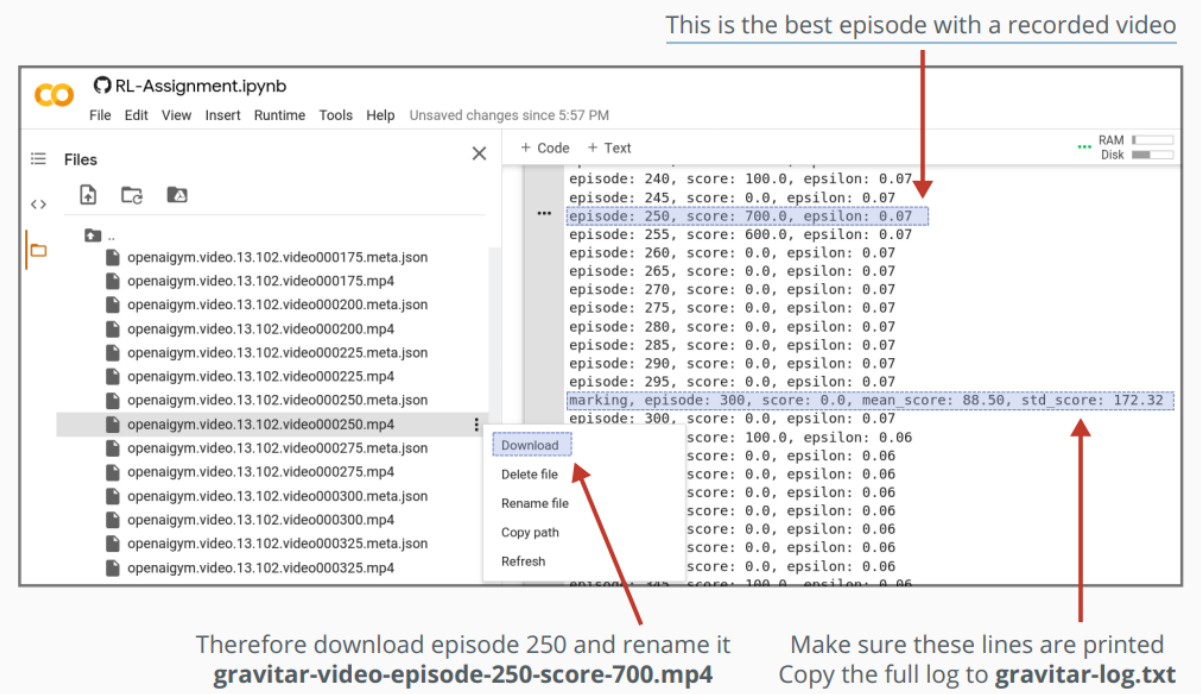
任务

您的任务是使用链接在首页上的入门代码来构建和训练深度强化学习代理，以获取最高的成绩。OpenAI 体育馆分别提供两个 Gravitar 环境“ Gravitar-ram-v0”和“ Gravitar-v0”。 您可以使用它们中的任何一个，但是要注意输入数组形状的大小，如此处 W-Gym (openai.com)和此处 W-Gym (openai.com)所述。

您可以使用自己喜欢的任何强化学习算法，包括讲座中未涵盖的算法。

提交

下图说明了提交方式和提交方式。您可以将所需的任何数据打印到日志中，但必须保留以“标记”开头的字符串。该字符串在每 100 个情节中打印分数的平均值和标准偏差。您不得更改此设置。提交的视频应显示您能够成功录制的最高分。录制视频的速度非常慢，因此，您可能希望每 25 集或更多的情节中有 3 个视频。增加此间隔可能会稍微改善训练效果，但是您可能会错过录制好的情节。



图中文字翻译：
This is the best episode with a recorded video
Therefore download episode 250 and rename it　gravitar-video-episode-250-score-700.mp4
Make sure these lines are printed　Copy the full log to gravitar-log.txt
这是录制视频最好的一集
因此，请下载第 250 集，并将其重命名为 gravitar-video-episode-250-score-700.mp4
确保这些行已打印将完整日志复制到 gravitar-log.txt

代码提交必须简洁明了。您可以提交.ipynb 或.py 文件。您无需遵循 PEP-8 或任何部门准则来进行此代码质量检查。尽量减少注释，因为好的代码应该说明一切。如果您尝试了许多令人印象深刻的设计和想法，则可以在提交内容的底部以较大的注释（或者，如果愿意，可以在.ipynb 中的文本）中进行简要说明，但不要期望这会对最终解决方案的标记产生重大影响。

防篡改

将对提交的日志文件进行统计分析。在 NCC 上使用 SLURM 脚本，将对收敛行为具有异常值的任何日志文件进行多次关联代码重新训练，以了解其预期收敛行为的分布。如果提交的日志数据的可能性表明存在篡改的迹象，将进行部门调查。如果有兴趣，此检测篡改的方法类似于解决 Zelda

Windwaker 中的 W Sploosh Kaboom 的部分解决方案。

打分

代码，视频和日志提交将标记为：

- [30 分]趋同和得分
  - 代理商的学习效率如何？
  - 经过大量的培训后，它是否会很幸运？
  - 多久获得一次好成绩？
- [10 分]解决方案的成熟度和适当性
  - 您将相关理论应用于该问题的程度如何？
  - 您的实现方式是多么骇人听闻，还是功能强大且设计合理？
  - 您刚刚引用并粘贴了代码，还是他们对进一步学习和新颖设计的理解力超出了教材范围？
- [10 分]视频的直觉
  - 视频看起来像特工在很多情节之后随机地变得幸运吗？
  - 是否有效地追捕敌人？
  - 是优雅地导航重力，还是像它由无限的可能性驱动？
  - 它是否学到了任何令人惊讶的行为？

PyTorch 培训

可以完全使用 Google Colab 完成此分配，或者您可能希望注册一个帐户并在 NCC 上进行培训。建议没有远程服务器经验或工作排队系统经验（例如 SLURM）的用户继续使用 Google Colab，这与 PyTorch 培训的速度一样快。如果使用 NCC，请仔细阅读文档并尊重工作排队系统上的其他用户。

结束语

我希望您喜欢这个课程。如果您在挣扎，请参加每周一次的变焦会议，询问问题并讨论任何问题，例如编程或相关理论。

问题与解答

强烈建议您仔细阅读这些常见问题和解答。

问：我已经花了一百万小时训练这些模型，但我仍然需要更多时间。这项作业是不可能的！
答：这一直是深度学习和强化学习的问题。我们的某些模型需要花费数月的时间进行训练，因此您需要非常仔细地选择实验，只有经过充分测试的代码才能证明您将在更多的时间收敛。而且，从某些方面讲，是的，这些任务不可能完美。这就是这些不确定的学习问题的本质，但是我并不是在寻求满分的完美。标记时，我将仔细查看该方法的预期收敛行为，而不是直接查看最佳分数/图像。

问：我刚刚读了这篇有趣的新论文，它与本课程中的技术有很大不同。我可以使用吗？
是的！这正是您应该做的事情。研究该论文并重新实现它，但确保在报告中引用它，并且还必须在代码顶部的注释中引用它。

问：我在网上找到了与我需要的代码相似的代码。我可以使用吗？
答：是的，但是您必须在书面报告和代码顶部的注释中都引用该代码。然后，我将在您的实现中仔细交叉引用该代码，以查看您对他们的代码的适应程度。如果您只是简单地复制和粘贴而没有经过实验或剪裁的证据，即使您的成绩很好，也不要指望获得良好的成绩。如果我在讲课中看到原始解释，新颖理解和理论应用的证据，即使实验结果不那么出色，您也可以得到很高的分数。但是，如果您假冒他人的代码，并“忘记”引用他们的代码，或者与其他学生一起工作，您很可能会陷入困境（请参阅 DUO 上的“抄袭和共谋”部分以了解用于检测到此）。这会招致非常严厉的部门处罚。

Q:手动图像编辑是什么意思？我可以编写代码来执行 X 吗？
A: 答案是“在合理范围内”。如果您在输入或输出上手动绘制机翼，您将得到零。如果您编写代码以手动设置看起来像机翼的像素，那么也会得到零。如果您开始使用 OpenCV 库来增强和增亮图像等，那么从技术上讲，这是可以接受的，但是不要期望为此获得额外的标记，因为这是深度学习作业，而不是计算机视觉或图像处理。

Q:复制最佳技术不是最佳策略吗？
A:实际上，这些任务的最新技术是大型分布式模型（使用大量水平缩放的 GPU）。您没有可用的这类资源，因此您将需要不同的策略。最先进的模型也经过了高度调整，可以将果汁中的每一滴都榨干。如果您直接复制此代码（如上一个答案所述），您将不会获得良好的成绩。另外，我将基于融合行为进行评分，而不仅仅是最终得分。如果我发现有证据证明质量差的模型已经在 AWS 上用大量的 GPU 进行了数千英镑的培训，则不会获得比仅在单个 Google Colab GPU 上进行了几个小时的培训的优秀模型更高的评分。 –即使分数更高。

Q:要获得 STL-10 奖励，我可以在 CIFAR-10 上进行训练，然后在 STL-10 上以全分辨率进行一个优化步骤，这样我可以说我已经在其中进行了训练吗？
A:这不会显着提高输出质量，因此我不会为此而增加收益。

Q:我是否需要训练所有数据？
A:不，您可以按照自己喜欢的任何方式训练 CIFAR-10 或 STL-10 的零件，或它们的组合。

Q:我有一个很棒/有趣/很棒的视频。分数虽然不高，但令人惊讶！我也可以提交吗？
A:是，您可以上传另外一个据此命名的精彩视频：gravitar-outtake-episode-100-score-50.mp4

Q:什么是预先训练的体重？
A:在这里您将使用别人训练过的不同数据模型，例如在 ImageNet 上训练的 GAN，然后为新任务转移权重（转移学习）。尽管这会生成漂亮的高分辨率有翼马，但不允许这样做。

Q:我可以使用 Keras，Theano，Tensorflow，Caffe，Chainer 或 X 来开发课程吗？
A:不会，除非您已经对 PyTorch 很有经验，并且希望使用此作业来学习另一个框架（例如 JAX）。在这种情况下，您必须给我发送电子邮件并提供证据，证明您已经拥有丰富的 PyTorch 经验，因为这是 DL 研究的行业标准。

Q:我真的很挣扎，感到不知所措。数学太复杂了，我不知道从哪里开始。
A:好的，首先不要惊慌，弄湿毛巾擦拭脸:)现在打开干净版本的 Colab Starter Code 并尝试运行它。接下来，尝试对模型进行一些小的修改。如果发现错误，请慢慢阅读，然后将其搜索。从尝试改进现有代码开始-为什么不尝试从一开始就使其残差？尝试感受一下它的局限性。当您有足够的信心时，请尝试实施一些您认为会做得更好的更复杂的事情。在这里，您需要小心一点，不要一次尝试太多。尝试添加尽可能少的附加功能，然后对其进行测试并检查其是否有效。逐步逐步建立，直到您有信心尝试更高级的想法。

Q:是否可以免费使用 Google Colab？
A:是的，尽管当 Google 承受沉重负担时，它们会对资源施加了限制（有所不同）。因此，要充分利用 Colab，请像示例代码中那样保存模型和优化参数，并在完成后关闭选项卡。这是因为 Google 优先考虑使用较少资源的用户。

Q:我需要 1 次 Gravitar 得分吗？飞马座能给我第一吗？
A:我无法回答，因为标记“功能”的参数太多。

Q:页面限制中是否包含参考？
A:不，您的参考资料部分可以从第 5 页开始，不会受到任何处罚。您可以有无限数量的参考。

Q:我必须使用 LATEX 模板作为纸张吗？
A:是的，这是出于在该领域撰写会议论文的经验，希望它将鼓励更慢，更简洁的写作，以及严格的数学和专业排版。

Q:帮助！我从未使用过 LATEX，也从未教过 LATEX!
A:登录 overleaf.com 并克隆纸张模板后（单击 overleaf.com 主页上的复制图标，然后编辑复制的版本），只需修改部分文本并按 Ctrl S 即可进行编译。如果您想实现特定目标，请使用 Google。然后，您可以直接从背面下载并提交已编译的.PDF。其余的自然而然。