**Learned Motion Matching**

Source: <https://static-wordpress.akamaized.net/montreal.ubisoft.com/wp-content/uploads/2020/07/09154101/Learned_Motion_Matching.pdf>

I had forgotten but there is a very good concise explanation in the website you sent me. I will link to it here.

Good explanation: <https://montreal.ubisoft.com/en/introducing-learned-motion-matching/>

Very Brief Explanation:

Learned Motion Matching is basically trying to replace an existing algorithm.

We want to replace the algorithmic characteristics by replacing it with a bunch of neural networks and hope it gives a much more efficient computation.

Indeed, as we will see the algorithm is improved upon drastically. Let’s start with presenting the original algorithm we are working to alter. It is *Motion Matching*.

**Motion Matching** is a method of searching a large database of animations for the animation which best fits the given context.

Here are several links which proved to be useful to me:

1. Link Provided by Roi:   
   <https://montreal.ubisoft.com/en/introducing-learned-motion-matching/>
2. Link to Paper:   
   <https://static-wordpress.akamaized.net/montreal.ubisoft.com/wp-content/uploads/2020/07/09154101/Learned_Motion_Matching.pdf>
3. Link to Supplementary Video (from Paper):  
   <https://www.youtube.com/watch?v=16CHDQK4W5k&feature=youtu.be&ab_channel=UbisoftLaForge>
4. Learned Motion Matching Webpage Official:  
   <http://theorangeduck.com/page/learned-motion-matching>

**Mode Adaptive Neural Networks for Quadruped Motion Control (SIGGRAPH 2018)**

This is the second paper we’ve read – we did not work on it. However, since I’ve read it, it might be a good idea to briefly explain the paper to facilitate whomever needs to work on it.

Note I: This is not enough to understand the paper of-course so if you’re reading this, make sure you look through the paper later as well.

Note II: Since the paper we worked with is a continuation of this one, there may be similarities between the two, however they are different papers and the other one has much more detail about its inner functions.

Github: <https://github.com/sebastianstarke/AI4Animation>

**Summary**

Introduction + System Overview: Quadruped motion is an important unsolved problem in computer animation. It’s importance is profound in computer games, films and also **robotics**. Animating quadruped motion is hard. There are many tedious and difficult to perform tasks (even for humans) that the animators/engineers must do. (See Paper for more details).

The goal will be to diminish the need for those tasks. In fact, we will be constructing motion in a data-driven fashion. The paper proposes Mode-Adaptive Neural Network (MANN), an novel NN architecture which can learn from quadruped motion.

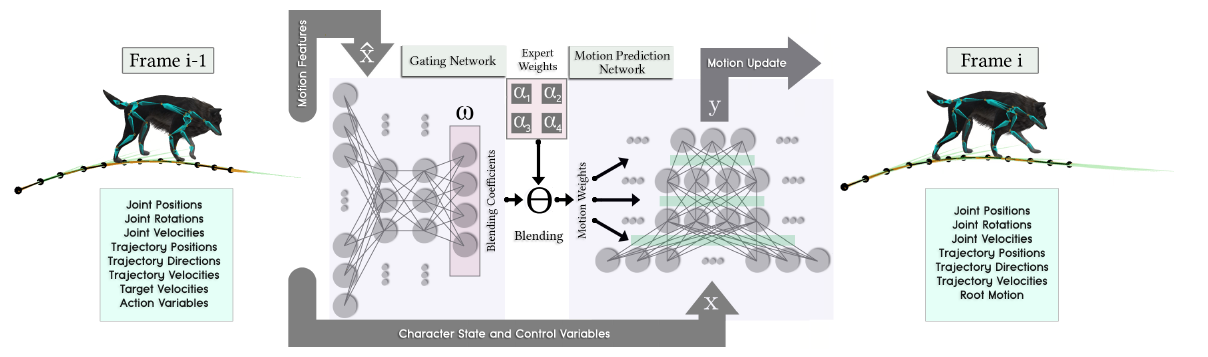
The system has: motion **prediction network**, and **gating network.**

At a very high-level:

For each frame: given the **state** in the previous frame and **user control** input, the network outputs the character new state (for the current frame).

More specifically (per frame), the **gating network** will dynamically update the **prediction network**’s weights. How? There are **expert weights** that are weights tuned for the **prediction network** to perform a specific action (animation). The **gating network** will take these expert weights and **according to context** it will interpolate the expert weights and embed it into the **prediction network**.

This is the general idea. See image below.



The left side (Frame i-1, Motion features, Character state and control variables) are input from the previous frame – the **state**.

The right side is the output of the iteration, (Motion Update).

System Overview: You are now ready to read this section on your own – it shouldn’t be difficult. 😊

Preprocessing (Data Preparation): You should read the paper itself for more details. However, the main thing to take first is:

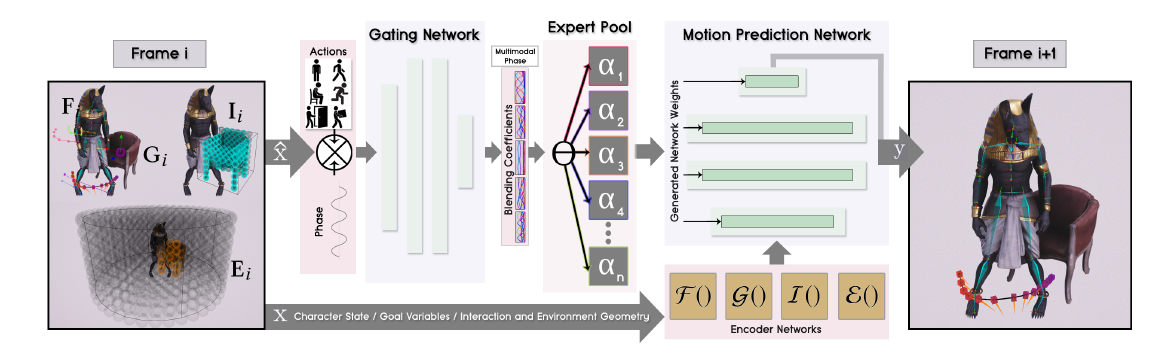
1. Motion Classification: the animations are classified into locomotions, sitting – standing – idling – jumping – lying. The classification is manually done.
2. Locomotion Modes: See Paper.

The rest of the paper explains the details in an easy-to-understand manner. I do not want to delve into the exact building of the neural network because there is a lot to explain and I wasn’t able to produce a much better job than they did in the paper. Good luck!

**Neural State Machine for Character-Scene Interactions**

Essentially, almost everything I said in the previous paper holds here too. (This is a continuation based on the previous paper).

Let us look at the architecture:



On the left: input same as before.

On the right: output same as before.

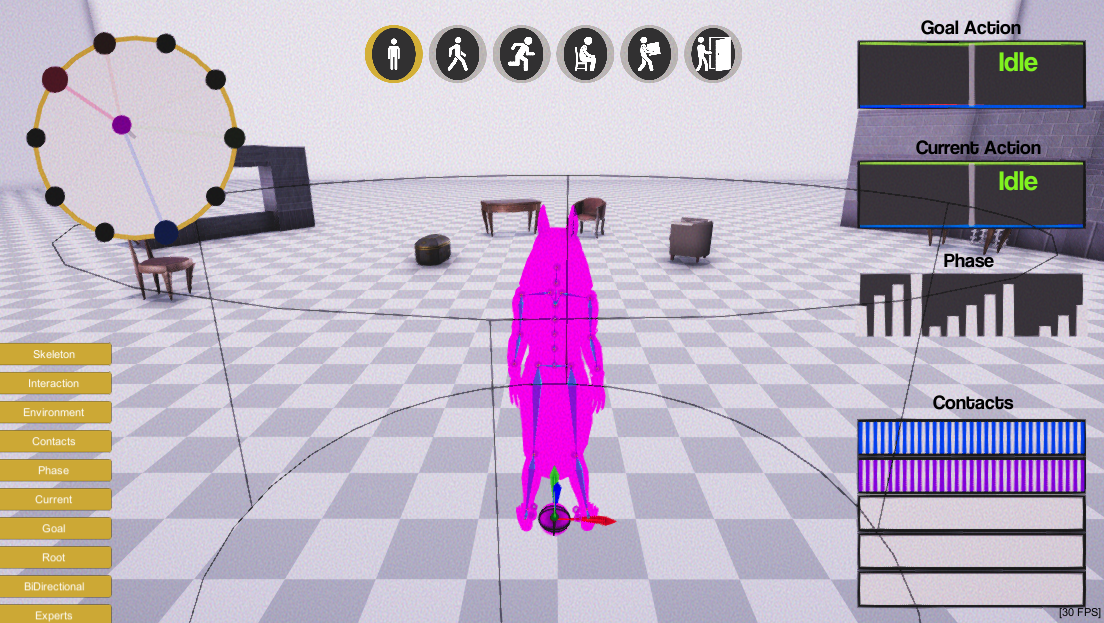
The Gating Network outputs the blending coefficients of the expert pool which ar applied to the Motion Prediction Network. Same as before.

The only difference is the Encoder Networks. These are simple three-layer networks which are networks that try to encode the environment and user input in some meaningful way (meaningful in the sense of the Motion Prediction Network itself). – See paper for explicit details about these encoders.

The rest of the paper is not that hard to understand.

I will provide **partial** explanation on the GUI of the project.

Once you hit play, you will see this screen:

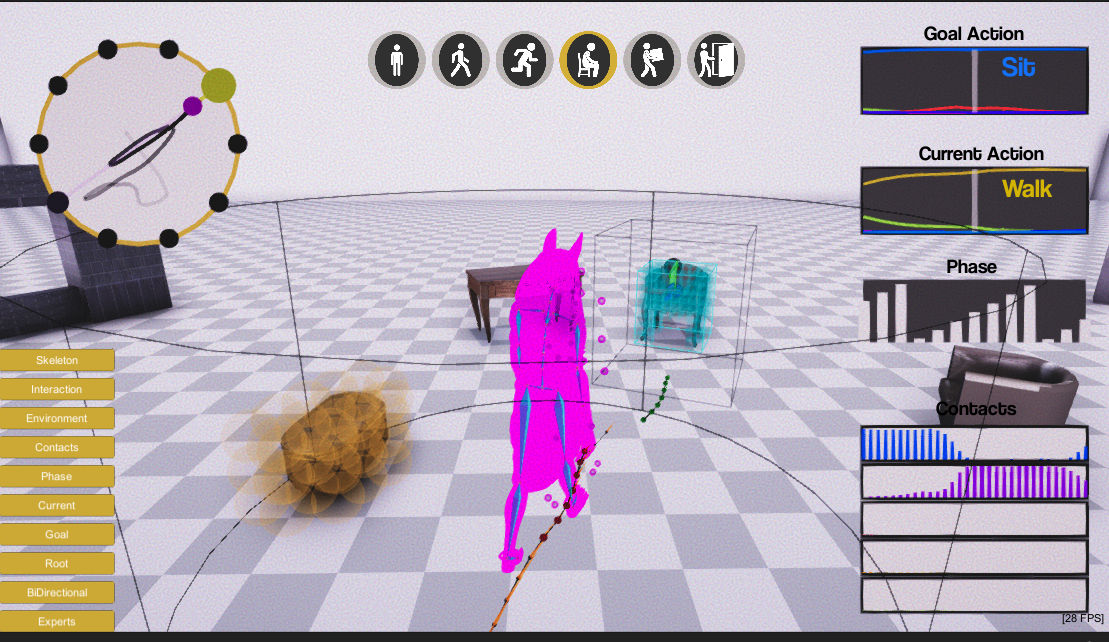


Everything on the lower-left is a toggle button. Play with those to understand what each of these represent. I will explain the most important ones here.

The “Experts button” will toggle the circle visible (invisible) when clicked. What that circle is representing is the current blending of the coefficients output by the **gating network**. You see 10 dots revolving around the circle, each dot represents an expert.

The number of experts is a hyper-parameter of the architecture and can be played with to suit your needs.

The “Goal Action” is the action that the system thinks you want to achieve. The “Current Action” is the action the character is currently performing. It’s good here to think about the encoder networks.



For example if you hit “C” on the chair you can see the goal action changes to “Sit” in the image above.

Personal Advice for future students on how to read the paper:

After understanding the high-level details provided here, you should first and foremost go interact with the project in Github. It will give a feel of how everything is working under wraps.

Then, read the paper once seeing that you can recognize all the details provided in this document. Once you see the connections, you should then move on to reading the paper more thoroughly. It will help set things right in how the whole system works.