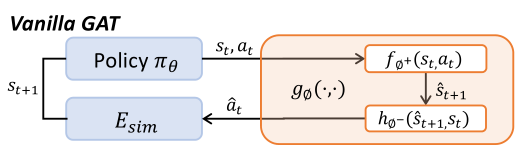
Grounded action transformation(GAT):

**Goal**: address the performance gap that arises when transferring policies from simulation to real-world scenarios.

**Key idea**: apply the learned policy to real-world data and minimize the discrepancy

**Limitation**: need substantial amount of real-world data to fit all situations.

**Why LLM helps:** human prior knowledge helps infer the behavior for unique conditions that hard to observe in reality

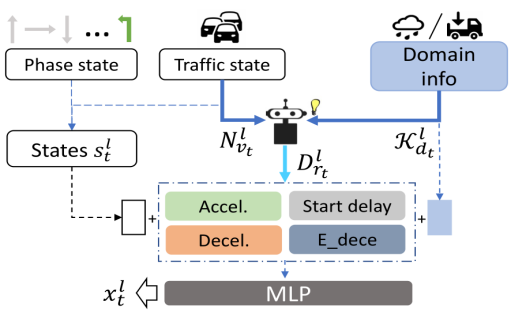
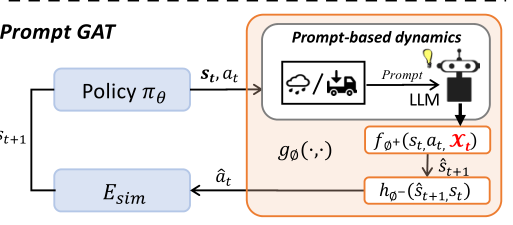


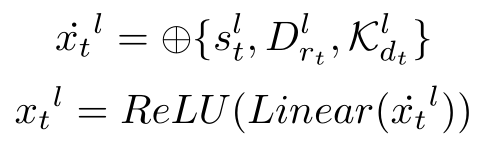
Forward model: wps, trained with **real** data to predict the next possible state

Inverse model: wps, trained with **sim** data to predict action that will produce C:/Users/JASONZ7/AppData/Local/Temp/wps.xnILipwps

Then, E\_sim will rectify C:/Users/JASONZ7/AppData/Local/Temp/wps.xnILipwps to wps so as to minimize the discrepancy.

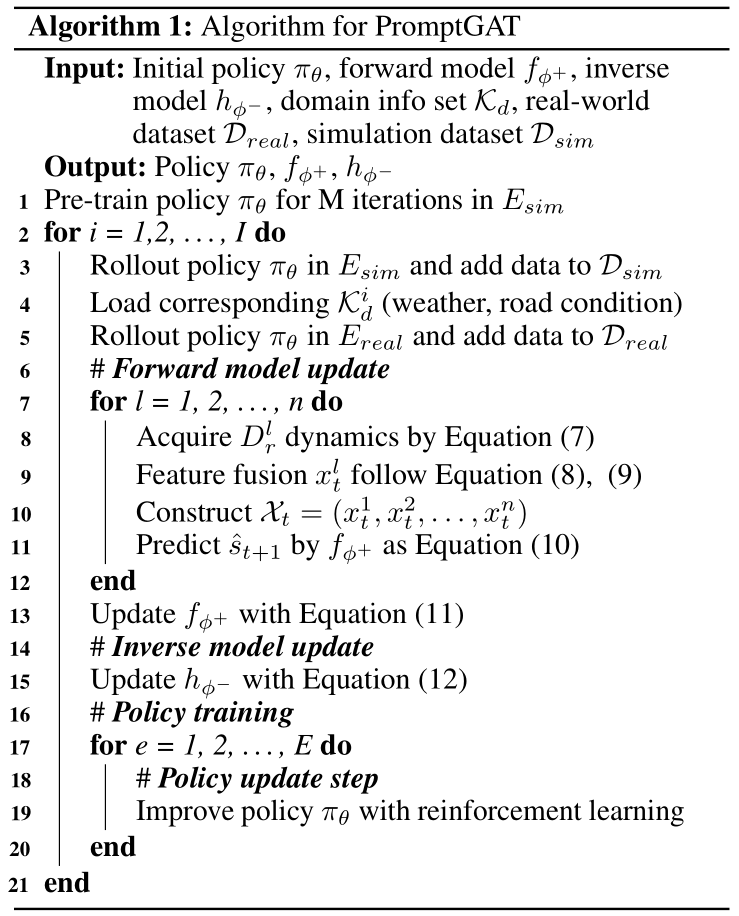
Sometimes, we may learn h well with large amount of sim data, but asymmetrical amount of real data, so maybe f is not learned very well. Therefore, extra information is expected to provide as inputs to f network, X\_t.



C:/Users/JASONZ7/AppData/Local/Temp/wps.BQAGNdwps, l is the lane index. wps is domain info: (weather, road), N is traffic state: e.g. how many cars on road. From these information, we use LLM to achieve the task: obtain the recommended (A, S, D, E). After that, three feature vectors are concatenated followed by and MLP to extract features..

**Loss** of forward model: C:/Users/JASONZ7/AppData/Local/Temp/wps.IUxJUTwps, s, a are collected from E\_real.

**Loss** of inverse model: Categorical cross-entropy loss: C:/Users/JASONZ7/AppData/Local/Temp/wps.rwKCwowps s, a collected from E\_sim.



LLaMA: (decoder-only)

Main difference:

1. Pre-normalization[GPT3]: normalize the input of each transformer sub-layer, not output LN
2. SwiGLU activation function
3. Rotary embeddings: relative positional embeddings

**Zero-shot**: a textual description (task) and a test example

**Few-shot**: a few examples of the task (examples+task description) and a test example

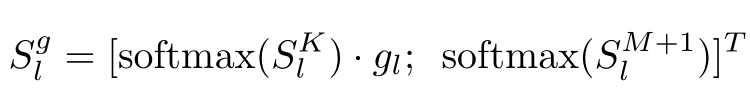
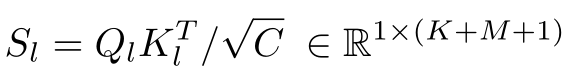
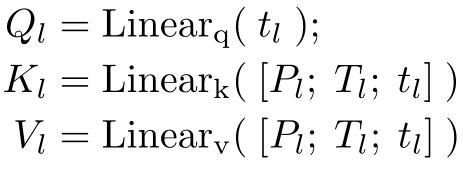
**LLaMA-adapter:**

Characteristics:

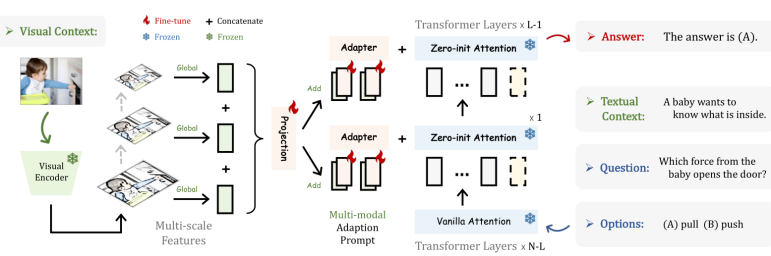
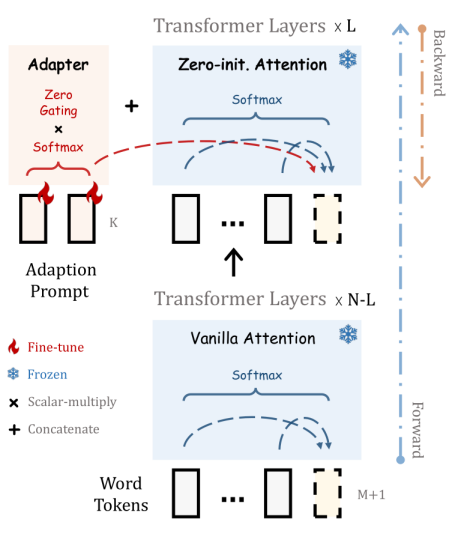
1. only 1.2M learnable paras upon frozen LLaMA 7B model
2. One-hour fine-tuning: zero-initialized attention
3. Multi-modal Instruction. Besides textual instruction, take images as input for multi-modal reasoning by adding image tokens into adaption prompts (image->prompts)

Learnable adaption prompts:

Trainable prompts for topmost (last) L layers with dimension wps, then concatenated with word tokens: 8d4a7f7a4f359bbde704c486e71b165. Frozen the first N-L layers’ parameters, and pass the input to obtain the prediction of the next token: t\_l. But this prediction is not good enough because no prompt-tuning. Therefore, we want to train a good prompt P that could produce a good output. A good way is to calculate attention of t\_l towards all other tokens. wps is also learnable, it is initialized to 0, which means prompts not effective (zero-initialized attention). For different heads, wps are also learned separately, benefiting the learning diversity of multi-head mechanisms (g <-> different prompts weight). But we still need to design target t\_l manually with soft prompt.



b76b4d734bf70d3aee21bd7e2b6a007



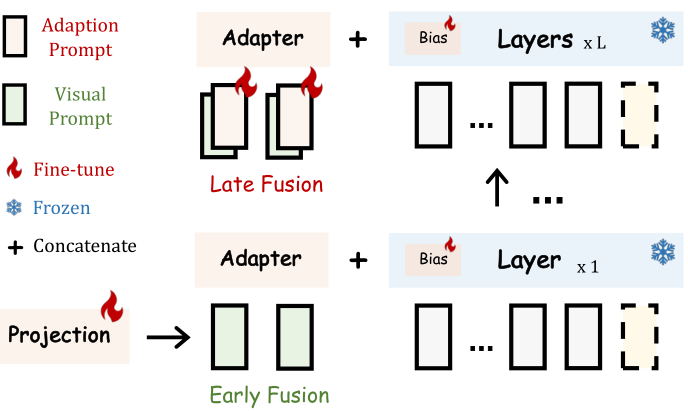
**LLaMA-adapter v2**: solve the problem: visual features dominate the adaptation prompts

Difference:

1. Unlock more trainable parameters(norm, bias, scale)
2. Early fusion strategy to feed visual tokens only into the early LLM layers
3. A joint training paradigm of image-text pairs and instruction-following data

In Llama-adapter, the input visual prompts are sequentially encoded by a frozen visual encoder with a learnable visual projection layer, and then added to the adaptation prompts at every inserted layer. In Llama-adapter V2, we instead inject the encoded visual tokens and adaptation prompts to different Transformer layers without fusing them together.

Here visual encoder is frozen, the projection from the global information to visual prompts is learnable.



Autotrain: build own dataset

Ludwig: T4 out of memory ->

Two reasons:

1. model parameters: <- quantization to 8bits

2. Adam optimizer <- LoRA(low-rank adaptation)

A diagram of a computer program

Description automatically generated

Next steps:

1. read <https://arxiv.org/abs/2307.08526>
2. fine-tune a caption model (e.g. llama2) to generate captions, using accident images labeled by safe, dangerous