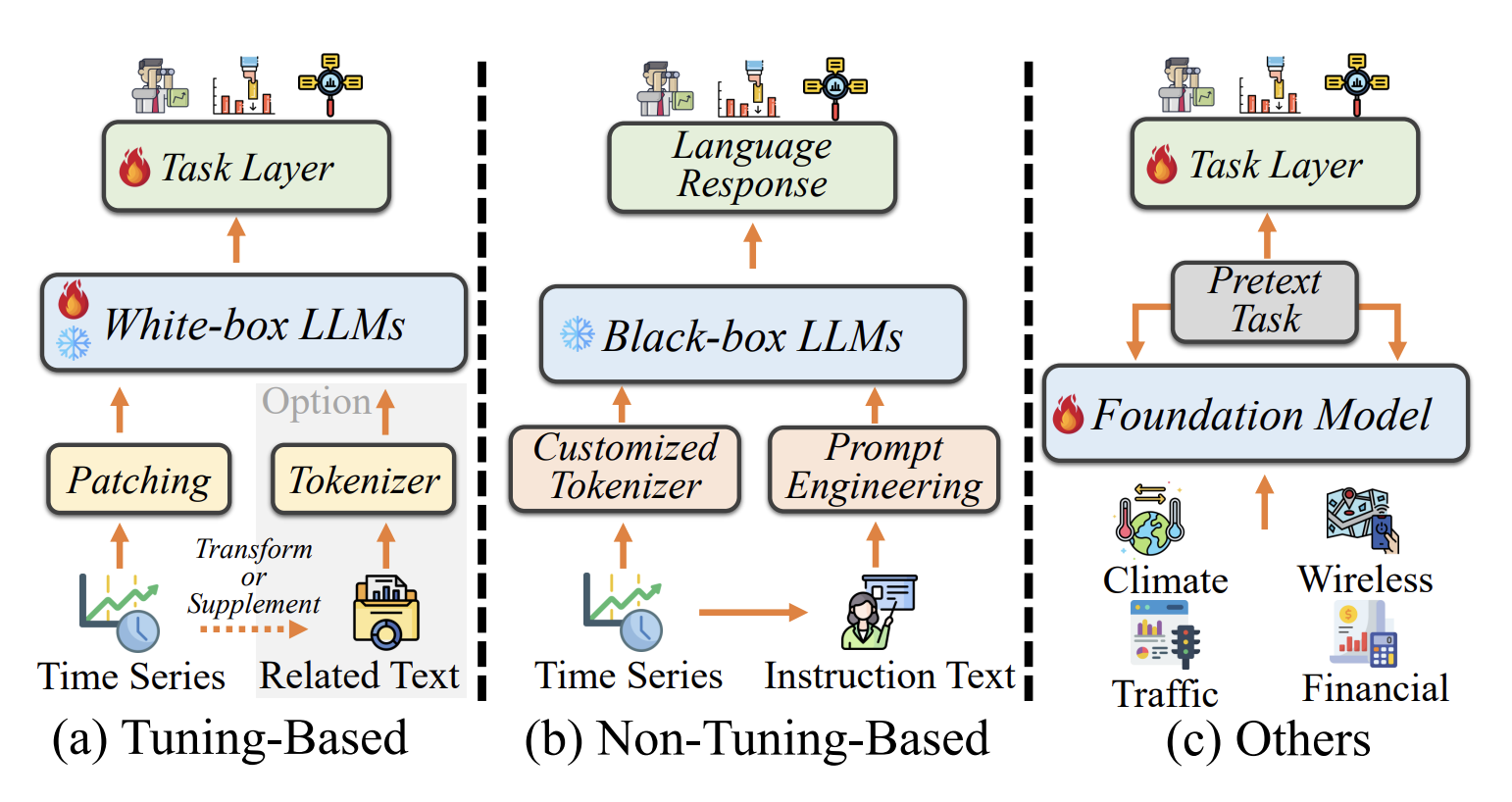
قضیه بزرگ تر از داده های مالیه مثلا اطلاعات سلامت ادما یا اطلاعات اب و هوایی

و خب همیشه ی سری عدد نی ممکنه ی سری اعداد باشن ک ب هم ربط دارن پس ممکنه گراف باشه

حالا روش های مختلفی وجود داره یک سری روش ها میشن fine tune بیس ها



که مهم ترینش time llm است

اولش ی ویدیو دیدم ازش نکته جالبش اینه اینا برخلاف مدلای قبلی که میومدن کل مدل را با داده های زمانی fine tune میکردند جلو نرفتند

اینطوریه که میان داده های زمانی رو تبدیل میکنن به یک مدلی و تکستی که مدل زبانی بفهمه

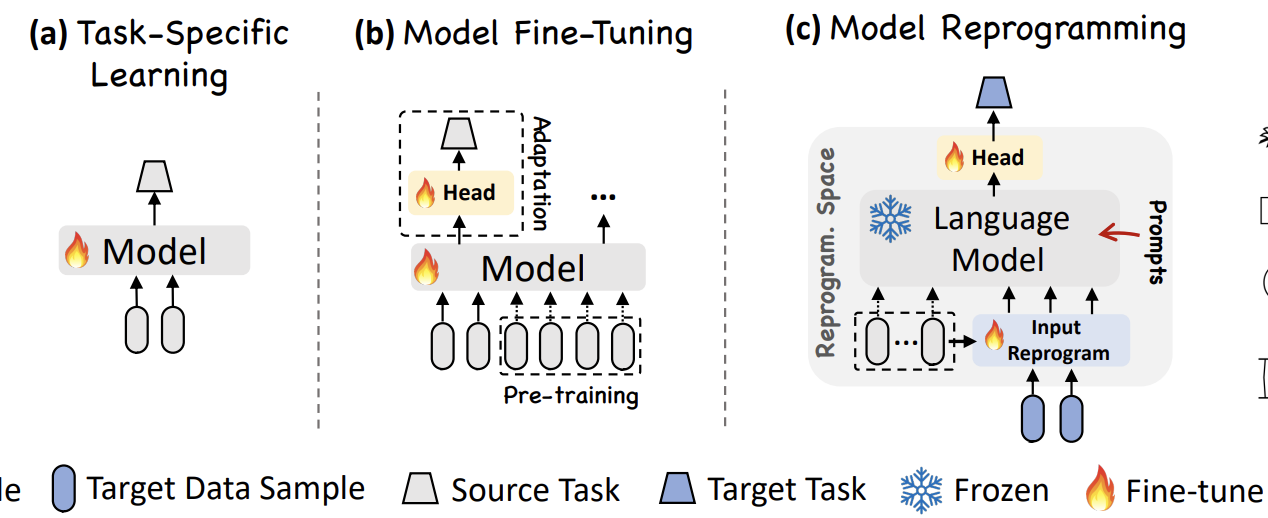
به همراه ی سری متن در مورد همون داده ها

و اینو میدن به مدل

بعد داده ای که مدل میده رو تبدیل میکنن به داده های زمانی

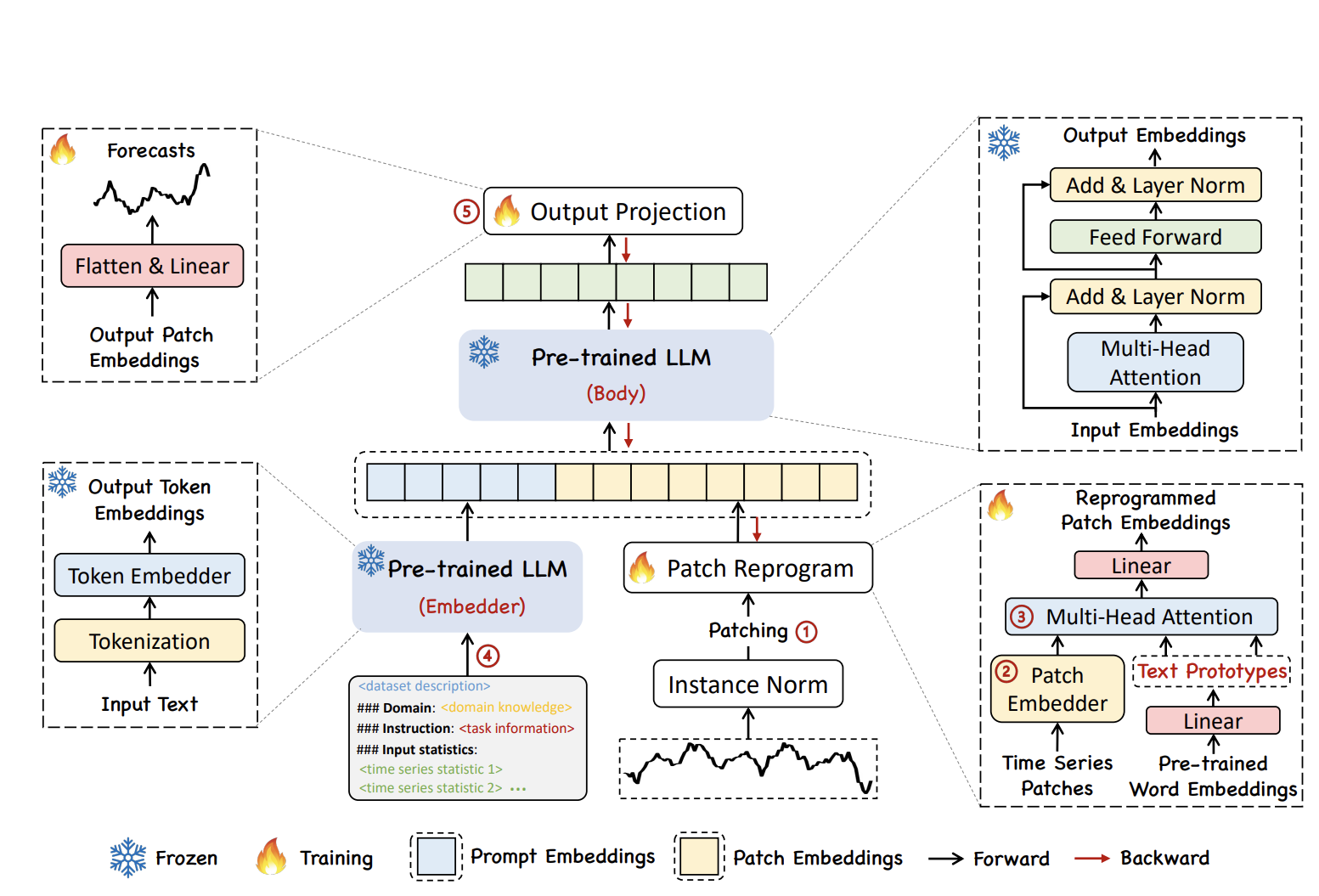
یک موضوعی که داره اینه که قضیه رو خیلی کلی نگاه کرده و محدود به یک نوع داده زمانی نی

و نکته بعدی اینه برای حالتی که zero shot یا few shot است هم خوب عمل میکنه میگ



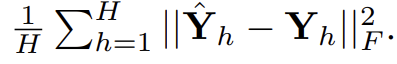
برای راه حل وسطیه اینطوریه ک مدل ها رو train میکنن با داده های زمانی و بعد برای یک تسک خاص fine tune میکنن

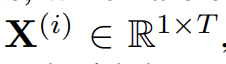
ولی مشکل اینه اونقدا داده نداریم



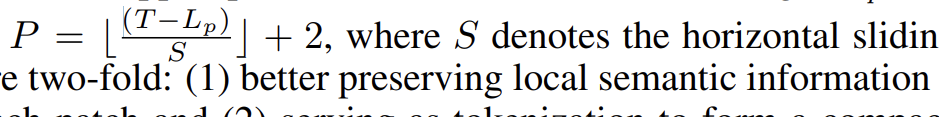
بردار وردی :  ما T تا تایم داریم هر تایم n تا ویژگی

بردار خروجی :  h تا بعدی رو تخمین میزنیم

 معیار هم خطا بدین صورتههه

 میشه ویژگی اول در تمام زمان

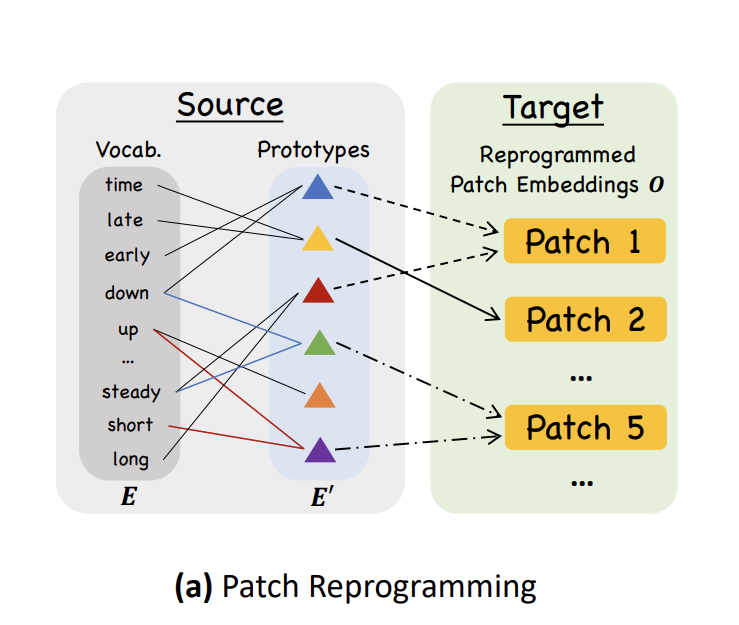
 خروجی اول در تمام زمان

میان xi رو اول نرمال سازی سازی میکنن یعنی میانگین صفر واریناس یک و بعد تبدیل میکنن به p تا patch   


طول هر کدوم lp هست و s هم گام افقی رو نشون میده --- > چون این پتچ ها ممکنه overlap داشته باشن

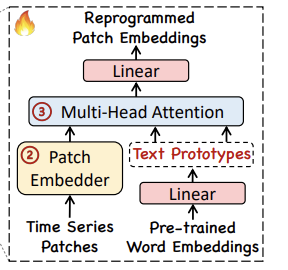
در اخر هم 

میایم lp رو میبریم به dm تا فقط ویژگی های مهم را داشته باشیم

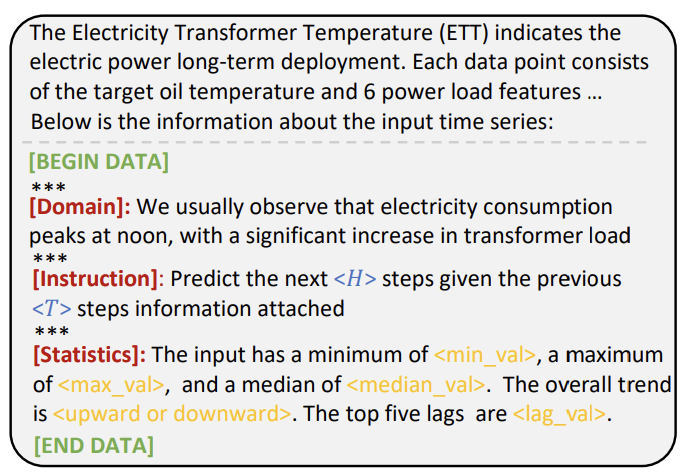


مسئله اینه ممکنه اونقدر این تبدیله lossless نباشه

قضیه اینه ک نمیشه ک از همه emdeding ها استفاده کرد ی سری proto type میاد مطرح میکنه اینو با اون داده های زمانی رو به عنوان key و value میده به توجه و بعد از ی linear رد میکنه



این مکانیزم توجهش با ی soft max



پس از دریافت نمایش‌های خروجی، ما ابتدا آن‌ها را فلت می‌کنیم، به این معنا که ابعاد ماتریس‌های نمایش خروجی را ترکیب می‌کنیم تا یک بردار یک بعدی بسازیم. سپس، با استفاده از یک پروژکشن خطی، این بردارهای یک بعدی را به نمایش‌های نهایی پیش‌بینی‌ها تبدیل می‌کنیم. این عمل به طور معمول با استفاده از یک لایه خطی در شبکه عصبی صورت می‌گیرد که وزن‌ها و بایاسهای آن به طور آموزش‌دیده تنظیم می‌شوند تا بهترین پیش‌بینی‌ها را ارائه دهند.

Tempo چه میکند؟

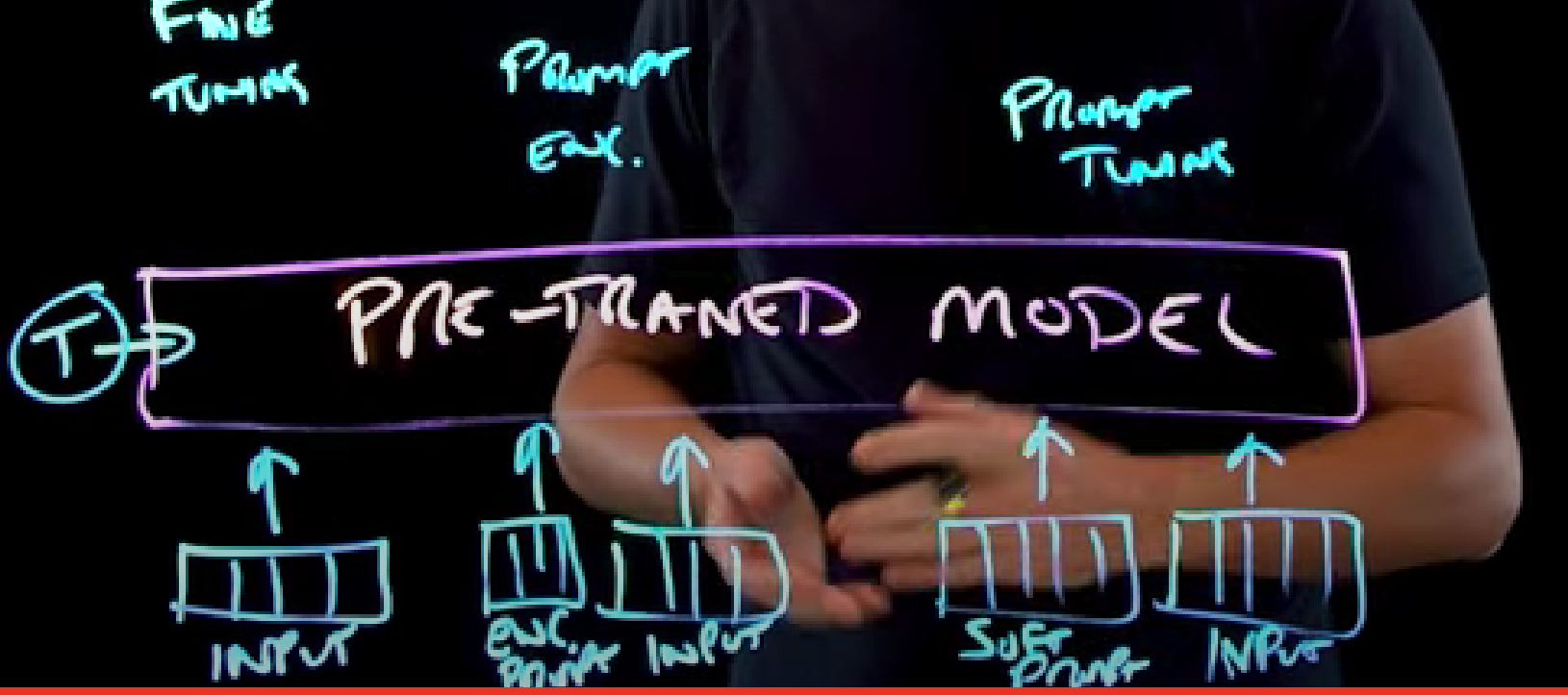
decomposition of the complex interaction between trend, seasonal and residual components;

introducing the selection-based prompts to facilitate distribution adaptation in non-stationary time series

یک قضیه روش خیلی تاکید داره اونم نوع و ساختار داده های زمانی است ک ی حال روند طور و فصلی دارند و مممکنه ی سری نویز داشته باشند

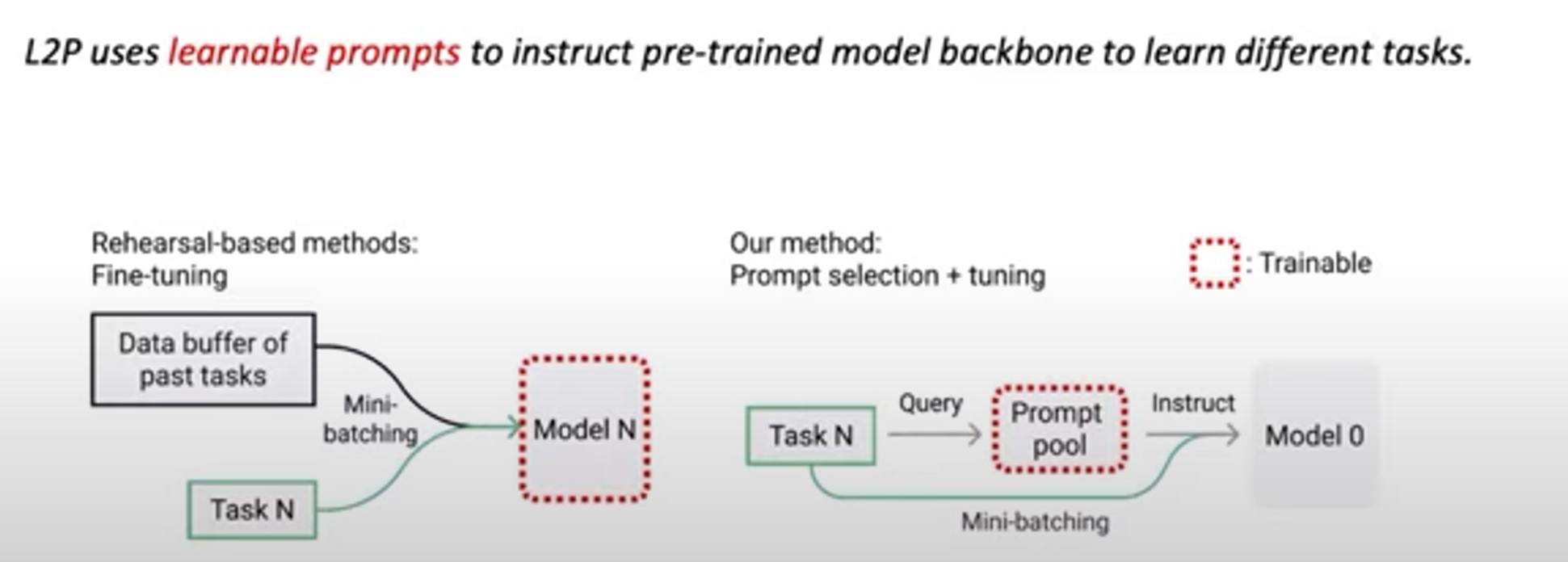
نکته بعدی اینه ک میخوایم general باشه ن فقد برای ی نوع داده زمانی

نکته بعدی prompt tuning



اینطوریه ک ما ی soft prompt داریم ک ai ساخته با ی input

بعدش راجب l2p میگ ک یکی از روش های prompt tuning



میگ نیاز ب buffer ندارم

میگ اومدم مشکل فراموشی رو حل کردم

یک موضوع جالبی ک میگ اینه ک ما اومدیم مدلمون رو از ترکیبی از روش های ml و روش های اماری و همچنین prompt pool ساختیم و زیبایی کار اینجاست و ی جورایی فرایند کانتینیوس لرنینگ دارن

خلاصه بعد این تو مقاله tempo یکم راجب این داده فصلی ترند و باقی مانده میگ ک انگار مجمئعش میشه کل داده های ما و خب حرفشون اینه ک اینکه ما میایم با stl اینا رو جدا میکنیم خیلی کار خوبیه چون transformer ها زیاد نمیتونن

داده های ترند ک میشن داده های بلند مدت فصلی میشه کوتاه مدت و باقی مانده میشه نویز اینا معمولا اینطوریه که میان ی فرایند اسموتینگ میانگین طوری رو داده ها میرن تا ترند رو به دست بیارن از بقیه داده ها فصلی رو پیدا میکنن باقی مونده هم میشه نویز

تحلیل آماری: تجزیه‌ی سری‌های زمانی به اجزای روند، فصلی و باقیمانده با الهام از روش STL برای درک ساختار زیربنایی داده‌ها.

یادگیری داده‌محور: انعطاف‌پذیری مدل‌های ترانسفورماتور پیش‌آموزش‌یافته برای تطبیق با الگوهای پیچیده در داده‌ها.

ترکیب این دو:

تجزیه‌ی روند و فصلی:

داده‌های خام پیش‌بینی به سه مؤلفه بر اساس روش STL تجزیه می‌شوند:

روند: تغییرات بلندمدت سری زمانی

فصلی: تغییرات تکراری وابسته به دوره‌های زمانی (فصلی، سالانه، و غیره)

باقیمانده: نویز باقیمانده بعد از در نظر گرفتن روند و فصلی

برای هر مؤلفه، میانگین‌گیری پنج نقطه‌ای برای کاهش نویز اعمال می‌شود.

هر یک از مؤلفه‌ها سپس معکوس‌سازی نرمال بر روی مقادیر انجام می‌شود (برای تسهیل انتقال دانش و کاهش خطاهای ناشی از تغییر توزیع) و در نهایت به فضای جاسازی پنهانِ مدل ترانسفورماتور نگاشت می‌شوند.

پچ کردن سری زمانی (Time-series patching):

مؤلفه‌های ترانسفورماتور شده به بخش‌های کوچک‌تری به نام پچ تقسیم می‌شوند. این کار به حفظ اطلاعات محلی در عین کاهش طول توالی کمک می‌کند.

کدگذاری زمانی (Temporal encoding):

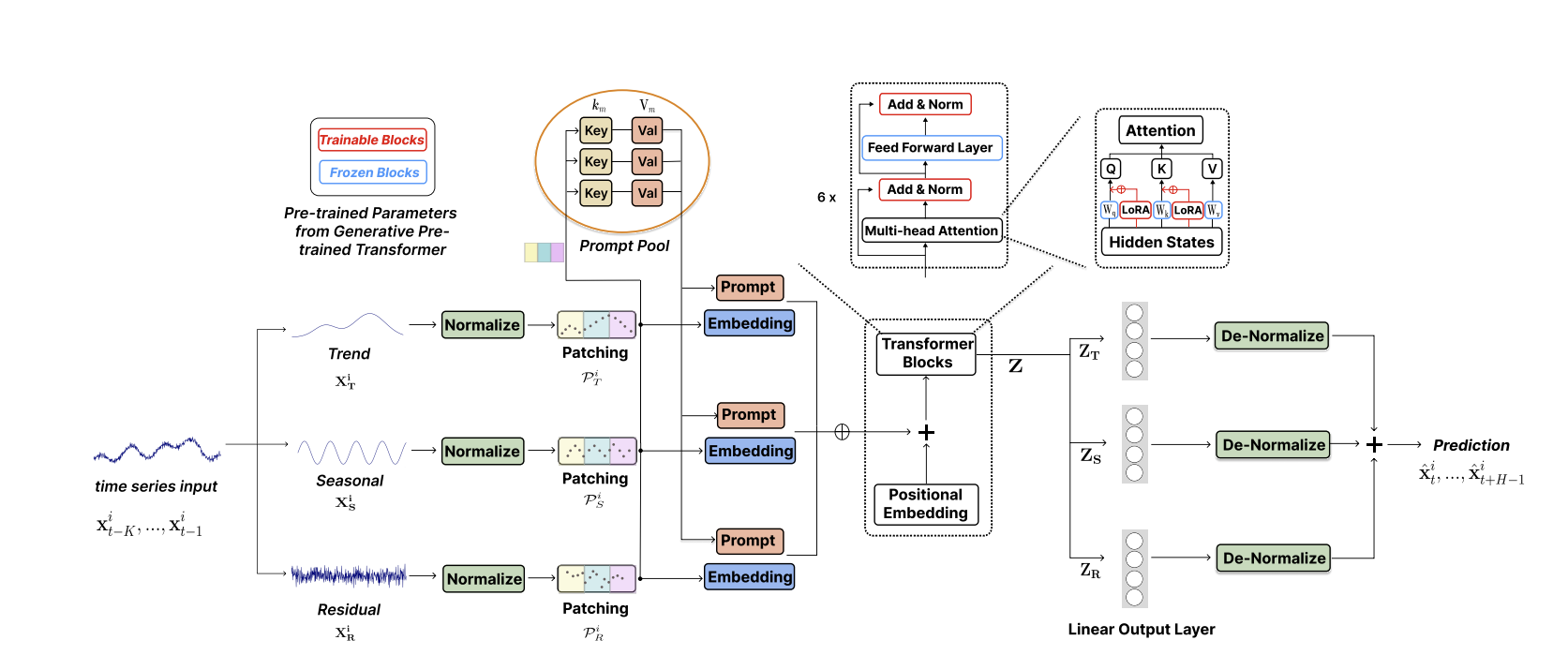
برای هر پچ، اطلاعات موقعیت زمانی آن نیز به عنوان جاسازی به مدل اضافه می‌شود تا ترتیب زمانی را در نظر بگیرد.

استخر Prompt (Prompt Pool):

این استخر مجموعه‌ای از Promptهای قابل آموزش را نگه می‌دارد که دانش زمانی را رمزگذاری می‌کنند.

در مواجهه با داده‌های جدید، مدل ابتدا این Promptها را با مؤلفه‌های ترانسفورماتور شده خودش مقایسه می‌کند و مشابه‌ترین Promptها را انتخاب می‌کند.

Promptهای انتخاب‌شده به عنوان راهنما برای پیش‌بینی جدید استفاده می‌شوند و به مدل کمک می‌کنند تا دانش مرتبط با گذشته را به درستی یادآوری کند.



مقاله test

It first tokenizes TS, builds an encoder to embed them by instancewise, feature-wise, and text-prototype-aligned contrast, and then creates prompts to make LLM more open to embeddings, and finally implements TS tasks

LLM-for-TS: در این رویکرد، به طور خاص برای داده‌های سری‌های زمانی، یک مدل بزرگ اساسی طراحی و پیش‌آموزش داده می‌شود، سپس مدل مورد نظر برای وظایف مختلف ارزیابی و تنظیم مجدد می‌شود.

TS-for-LLM: در این رویکرد، بر اساس مدل‌های زبانی بزرگ پیش‌آموزش داده شده موجود، آنها را قادر به کار با داده‌های سری‌های زمانی و وظایف آنها می‌سازد. به جای ایجاد یک مدل جدید، مکانیسم‌هایی طراحی می‌شود تا داده‌های سری زمانی را برای مدل‌های زبانی بزرگ سفارشی سازی کند.

رویکرد 1: LLM-برای-TS:

این روش به طراحی و پیش‌آموزش یک مدل جدید برای داده‌های TS نیاز دارد.

مزیت: راه‌حل بنیادی‌تری است زیرا پیش‌آموزش مرحله‌ی مهمی برای انتقال دانش به مدل است.

معایب: نیازمند داده‌های فراوان برای پیش‌آموزش، به دست آوردن چنین حجم و تنوعی از داده‌های TS دشوار است (خصوصا در بخش‌های غیرصنعتی)، و محدود به توانایی‌های نهایی مدل است.

رویکرد 2: TS-برای-LLM:

این روش بر سازگار کردن LLMهای موجود با داده‌های TS تمرکز می‌کند.

مزایا: نیازی به پیش‌آموزش گسترده ندارد، از داده‌های کمتر و راحت‌تری استفاده می‌کند، و می‌تواند از توانایی‌های موجود LLM استفاده کند.

معایب: ممکن است به دقت نهایی مشابه رویکرد 1 نرسد

LLM-برای-TS:

بیشتر برای متخصصان مناسب است.

قابل تعمیم نیست و نیازمند تخصص در حوزه‌ی خاص خود است.

روش ساده:

داده‌های سری زمانی را به توکن‌های متنی تبدیل می‌کند و آن‌ها را به صورت تک تک وارد LLM می‌کند.

معایب:

به دلیل ترتیب ورودی و چگونگی اتصال جملات، نتایج متفاوتی به دست می‌دهد.

LLM نمی‌تواند روابط بین توکن‌های مختلف را به خوبی بیاموزد.

اطلاعات با اهمیت که در وابستگی چند متغیره است، نادیده گرفته می‌شود.

روش پیشنهادی (جاسازی):

داده‌های سری زمانی را توکنیزه می‌کند.

مدلی طراحی می‌کند که این توکن‌ها را امبدینگ کند.

امبدینگ به دست آمده را جایگزین لایه‌ی امبدینگ LLM می‌کند.

مزایا:

مدل می‌تواند روابط بین توکن‌ها را یاد بگیرد.

اطلاعات موجود در داده‌های چند متغیره حفظ می‌شود.

کارآمدتر از روش ساده است.

Self-supervised contrastive learning :

در این روش، دو نمونه سری‌زمانی مشابه به هم نزدیک و نمونه‌های غیرمشابه از هم دور می‌شوند. هدف این است که مدل بتواند ویژگی‌های مشترک در نمونه‌های مشابه را تشخیص دهد.

خوبی ها:

یادگیری متضاد خود-نظارتی روشی مفید برای جاسازی سری‌زمانی بدون نیاز به برچسب‌گذاری است.

روش‌های تمایز نمونه‌ای در سطوح مختلف عملکرد قابل قبولی داشته‌اند.

بدی ها:

ارزیابی با مدل‌های ساده‌ای مانند SVM نمی‌تواند عملکرد واقعی در LLM را منعکس کند.

هماهنگی امبدینگ با فضای شناختی LLM برای استفاده موثر از آن حیاتی است

از text embedding ها میاد ب عنوان نمونه های اولیه استفاده میکند

نمونه‌های اولیه به عنوان لنگر یا راهنما برای نمایش داده‌های سری زمانی (TS) عمل می‌کنند.

مدل یاد می‌گیرد که نقاط داده TS را به گونه‌ای نمایش دهد که به نمونه‌های اولیه مربوطه در فضای نمایش "نزدیک" باشد.

این امر به هم‌ترازی نمایش TS با فضای شناختی و درک LLM کمک می‌کن

instance-level contrast

روش‌های رایج از تمایز در سطح نمونه استفاده می‌کنند. یعنی هر سری زمانی یا قطعه‌ای از آن، به طور مستقل در نظر گرفته می‌شود.

این روش برای مدل‌سازی وابستگی‌های پیچیده بین نقاط مختلف سری زمانی محدودیت دارد.

پیشرفت‌های جدید:

temporal-level contrast

در نظر گرفتن الگوهای زمانی مشابه و متفاوت برای تمایز نمونه‌ها.

این روش می‌تواند وابستگی‌های دوربرد را بهتر مدل‌سازی کند.

تمایز در سطح نمونه اولیه:

استفاده از مجموعه نمونه‌های منتخب ("نمونه‌های اولیه") برای نمایندگی الگوهای مختلف در داده‌ها.

مدل یاد می‌گیرد که سری‌های جدید را بر اساس شباهت آن‌ها به «نمونه‌های اولیه» در فضای جاسازی قرار دهد.

این روش می‌تواند الگوهای پیچیده‌تری را شناسایی کند.

تمایز در سطح نمونه اولیه (prototype-level contrast): استفاده از نمونه‌های منتخب برای نمایندگی الگوهای مختلف در داده‌ها.

در اینجا، نمونه‌های اولیه متن (textual prototypes) از LLM از پیش‌آموزشی استخراج می‌شوند تا فرآیند یادگیری را هدایت کنند.

منبع: الهام گرفته از Caron et al. (2020).

تفاوت‌ها:

روش‌های مختلف در نحوه تعریف نمونه‌های مشابه و نامشابه با هم تفاوت دارند.

برخی از روش‌ها (مانند تمایز در سطح نمونه) بر الگوهای محلی تمرکز می‌کنند، در حالی که سایر روش‌ها (مانند تمایز در سطح زمانی) بر وابستگی‌های دوربرد تمرکز می‌کنند.

برخی از روش‌ها (مانند تمایز در سطح نمونه اولیه) از اطلاعات پیش‌زمینه برای هدایت فرآیند یادگیری استفاده می‌کنند.

موضوع: توکنیزه کردن سری زمانی (TS) با استفاده از تابع Fs(x).

توضیح: این بخش اولین مرحله را توضیح می‌دهد که در آن سری زمانی به توکن‌های جداگانه تقسیم می‌شود.

روش رایج: استفاده از یک پنجره لغزنده برای قطعه‌بندی سری زمانی که در منابعی مانند Yue et al. (2022) توضیح داده شده است.

تعریف نمونه‌های مثبت (positives):

دو شیوه برای یافتن نمونه‌های مثبت برای هر نمونه اصلی پیشنهاد می‌شود:

1. نمونه همپوشانی: استفاده از توکن‌هایی که زیرمجموعه‌ای از آنها مشابه نمونه اصلی است.

2. نمونه تقویت‌شده: ایجاد دو نسخه تقویت‌شده از نمونه اصلی با تغییرات تصادفی برای بررسی حساسیت و مقاومت مدل.

روش‌های ایجاد نمونه تقویت‌شده:

ضعیف (jitter-and-scale): اضافه کردن نویز و تغییر مقیاس به سیگنال.

قوی (permutation-and-jitter): تکه‌تکه کردن توالی و به هم زدن تصادفی، برای بررسی تغییرات ساختاری.

تعریف نمونه‌های منفی (negatives):

نمونه‌های منفی توکن‌هایی هستند که هیچ زیرمجموعه‌ای مشابه نمونه اصلی ندارند.

بعد یک Fe داریم که exponentially dilated causal convolution network هست



اطمینان از نمایندگی مناسب embeding: با به حداقل رساندن تابع اتلاف خودرمزگذاری، کدکننده تشویق می‌شود تا embeding ihddرا تولید کند که اطلاعات کلیدی توکن‌های اصلی را حفظ کند. این امر به بهبود کیفیت و معنی‌داری embeding ها کمک می‌کند.

Bloomberge gpt

ی مدل گنده ی خروار پول خرج کردن نه دیتاست رو پابلیش کردن ن مدل رو بر اساس مدل bloom