TEMPORAL FUSION TRANSFORMERS FOR INTERPRETABLE MULTI-HORIZON TIME SERIES FORECASTING

2021.03.28

백지운

목차

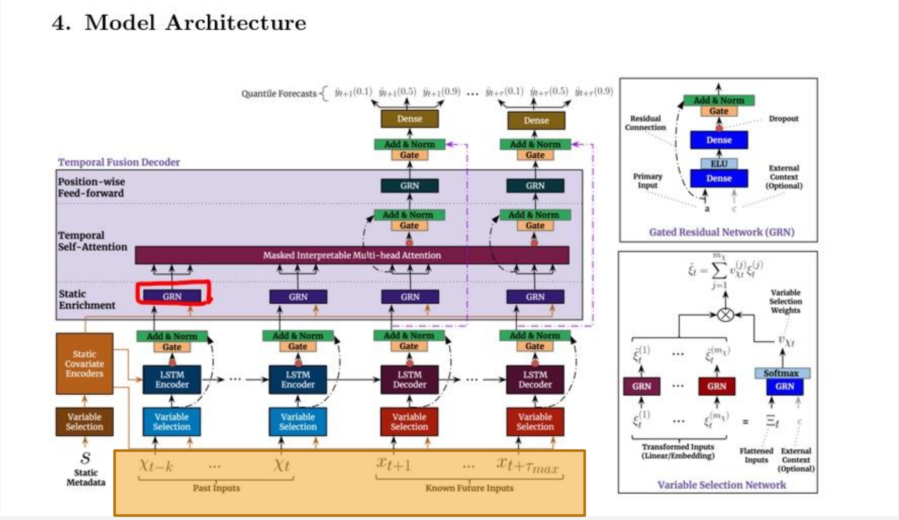
1. 연구 의의, 목적 등
2. 용어 정리
3. 모델 구조
4. Loss function
5. 데이터 셋 / 실험 결과
6. Interpretability
7. 결론
8. 코드

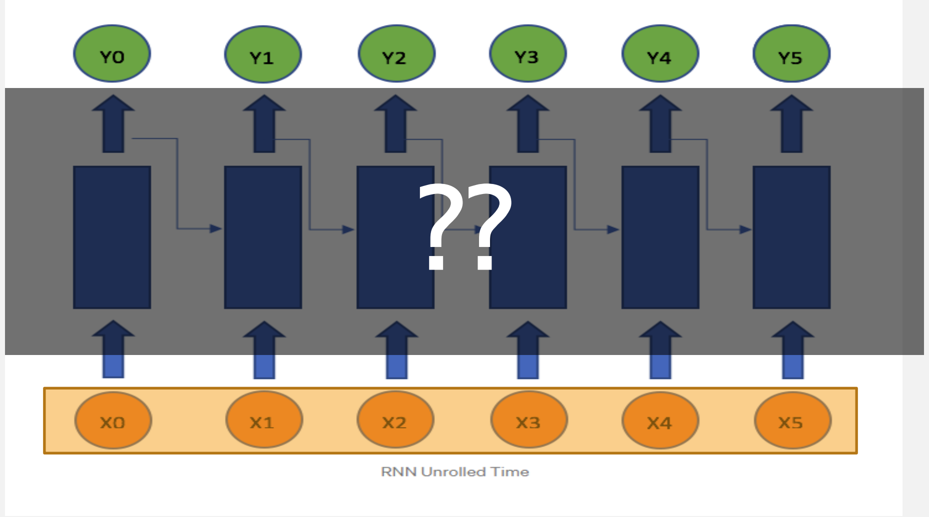
1. 연구 의의 및 목적

TFT VS RNN

ARCHITECTURE > GATING MECHANISMS
VARIABLE SELECTION NETWORKS
STATIC COVARIATE ENCODERS

ARCHITECTURE > GATING MECHANISMS ✕
VARIABLE SELECTION NETWORKS. ✕
STATIC COVARIATE ENCODERS ✕





input > Static covariates (contexts)
Observed inputs
Known inputs

input >
Observed inputs

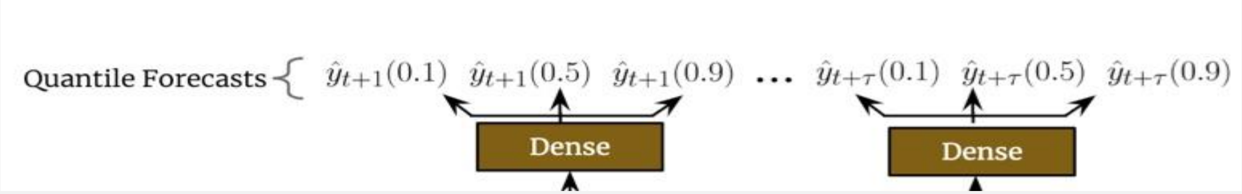
연구 목적

- Forecasting 에 영향을 줄 수 있는 보다 유연하고 풍부한 데이터를 모두 활용할 수 있는 모델을 만들겠다
- 모델 forecasting 도중 해당 시점의 연산에서 필수적인 레이어와 features 만을 필터링하여 사용하겠다
- Multi-head attention 의 변형 방식으로 다양한 헤드를 앙상블 느낌으로 각 타임스텝의 관계성을 폭넓게 해석하겠다 (interpretability)

2. 용어 정리

용어 정리

- Horizon : 예측 범위 / Multi- Horizon : 여러 개의 예측 범위



t가 월요일이라고 한다면 화요일 예측 값 월요일 + t 시점의 예측 값

- Static (=time invariant) covariates : 독립 변수가 종속 변수에 미치는 효과에 영향을 줄 수 있는 변수 ex. A 수학 문제집과 수학 성적과의 관계에서 학생들의 원 수학 실력 => **메타데이터**
- Observed inputs (z), known inputs (x) ex. The way of week at time t

용어 정리

Horizon : 예측 범위 / Multi- Horizon : 여러 개의 예측 범위

Static (=time invariant) covariates : 독립 변수가 종속 변수에 미치는 효과에

영향을 줄 수 있는 변수 ex. A 수학 문제집과 수학

성적과의 관계에서 학생들의 원 수학 실력

=>메타데이터

Observed inputs (z), known inputs (x)

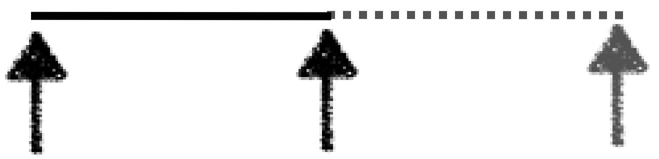
ex. The way of week at time t

$S_i \in \mathbb{R}^{m_s}$ Static covariates

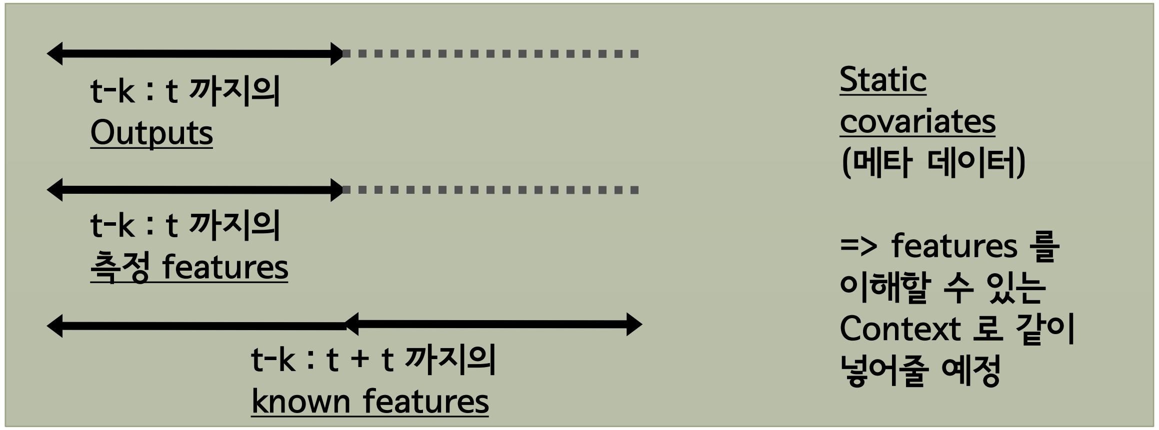
$\chi_{i,t} = [z_{i,t}^T, x_{i,t}^T]$ Inputs ; (observed, known)

$y_{i,t} \in \mathbb{R}$ Outputs

$$\hat{y}_i(q,t,\tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i)$$



이용할 variables 4가지



3. 모델 구조

모델 핵심 요소 6가지

Gating Mechanisms

모델 구조

자유도 크게

자유도 크게

자유도 작게

자유도 작게

자유도 크게

자유도 작게

=> 정제된 features

Variable Selection Networks

S,X,Y 중 각 시점에 꼭 필요한 features 필터링

Static Covariate Encoders

S 메타 데이터를 features 를 이해할 수 있는 context화

Interpretable Multi-Head Attention

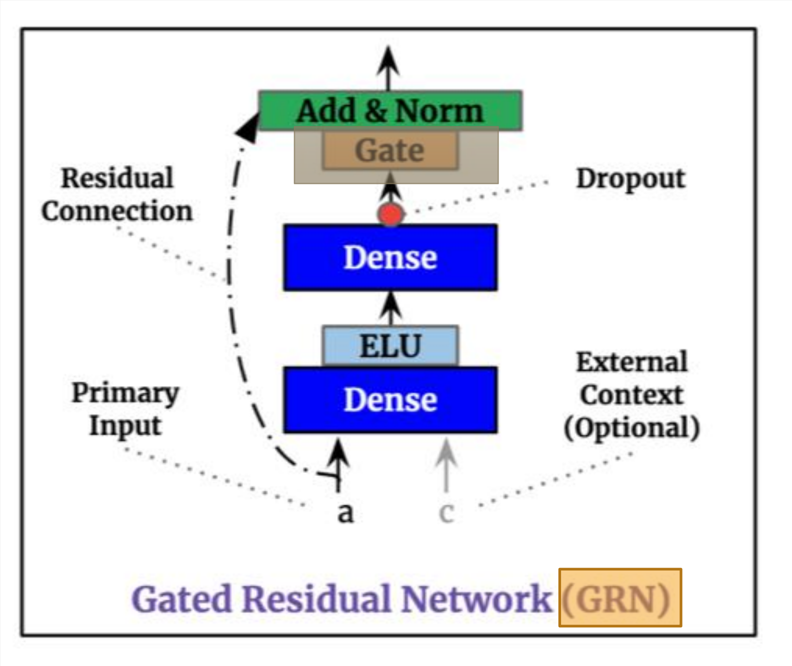
Temporal Fusion Decoder

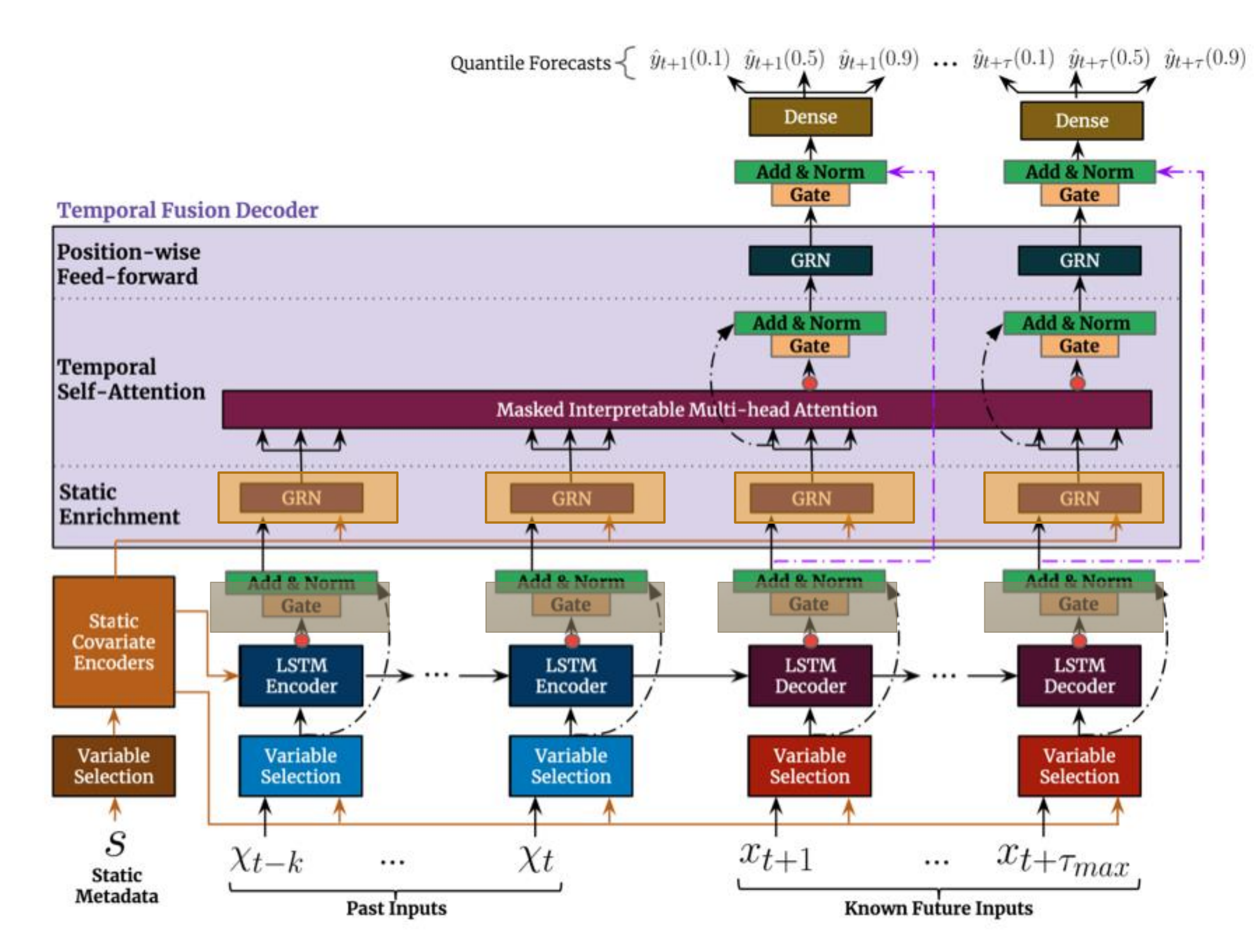
각 time step 의 장기간 상호관계 도출

Quantile Outputs

Gating Mechanisms = GRN layer

Gate 는 TFT 모델의 거의 모든 층에 사용 되는 핵심 테크닉
GRN layer 에도 gate 가 사용됨 !

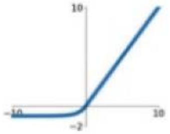




Gating Mechanisms = GRN layer

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



$$\eta_2 = \mathrm{ELU}\left(\boldsymbol{W}_{2,\omega}\, \boldsymbol{a} + \boldsymbol{W}_{3,\omega}\, \boldsymbol{c} + \boldsymbol{b}_{2,\omega}\right), \tag{4}$$

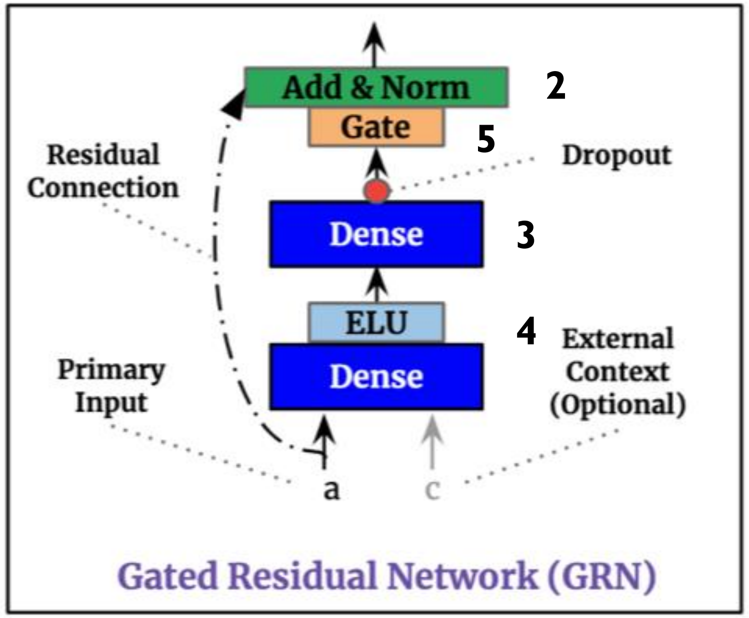
$$\eta_1 = \boldsymbol{W}_{1,\omega}\, \eta_2 + \boldsymbol{b}_{1,\omega}, \tag{3}$$

Gate

$$\mathrm{GLU}_\omega(\gamma) = \sigma(\boldsymbol{W}_{4,\omega}\, \gamma + \boldsymbol{b}_{4,\omega}) \odot (\boldsymbol{W}_{5,\omega}\, \gamma + \boldsymbol{b}_{5,\omega}), \tag{5}$$

Dropout (training)

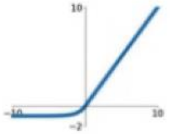
$$\mathrm{GRN}_\omega\left(\boldsymbol{a}, \boldsymbol{c}\right) = \mathrm{LayerNorm}\left(\boldsymbol{a} + \mathrm{GLU}_\omega(\eta_1)\right), \tag{2}$$



Gating Mechanisms = GRN layer

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



$$\boldsymbol{\eta}_2 = \mathrm{ELU}\left(\boldsymbol{W}_{2,\omega}\, \boldsymbol{a} + \boldsymbol{W}_{3,\omega}\, \boldsymbol{c} + \boldsymbol{b}_{2,\omega}\right), \tag{4}$$

$$\boldsymbol{\eta}_1 = \boldsymbol{W}_{1,\omega}\, \boldsymbol{\eta}_2 + \boldsymbol{b}_{1,\omega}, \tag{3}$$

$$\mathrm{GLU}_\omega(\boldsymbol{\gamma}) = \sigma(\boldsymbol{W}_{4,\omega}\, \boldsymbol{\gamma} + \boldsymbol{b}_{4,\omega}) \odot (\boldsymbol{W}_{5,\omega}\, \boldsymbol{\gamma} + \boldsymbol{b}_{5,\omega}), \tag{5}$$

Gate

$$\mathrm{GRN}_\omega(\boldsymbol{a}, \boldsymbol{c}) = \mathrm{LayerNorm}\left(\boldsymbol{a} + \mathrm{GLU}_\omega(\boldsymbol{\eta}_1)\right), \tag{2}$$

(4)

>= 0 ?

W (4) + b (3)

W (3) + b (5)-2 * (5)-1 (5)

(4)

< 0 ?

W(αe^ (4) -1) + b (3)

W (3) + b (5)-2 * (5)-1 (5)

Final Output (2)

LN(a+ (5))

LN(a+ (5))

LN(a+ (5))

...

Depends on (5)-1

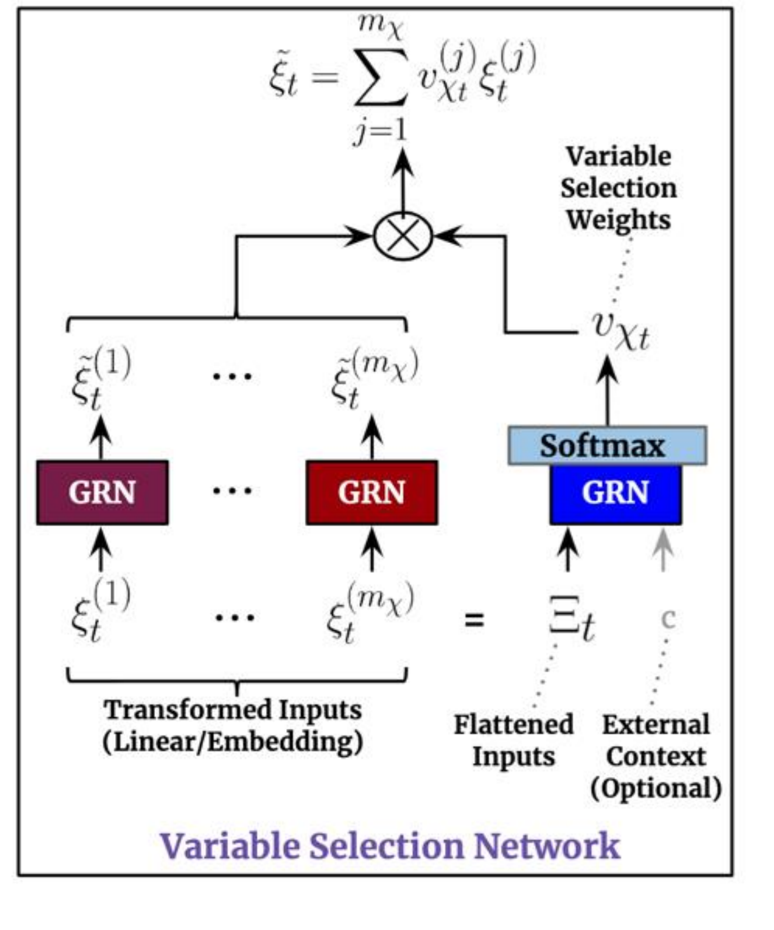
(4) 가 모델 복잡도 결정

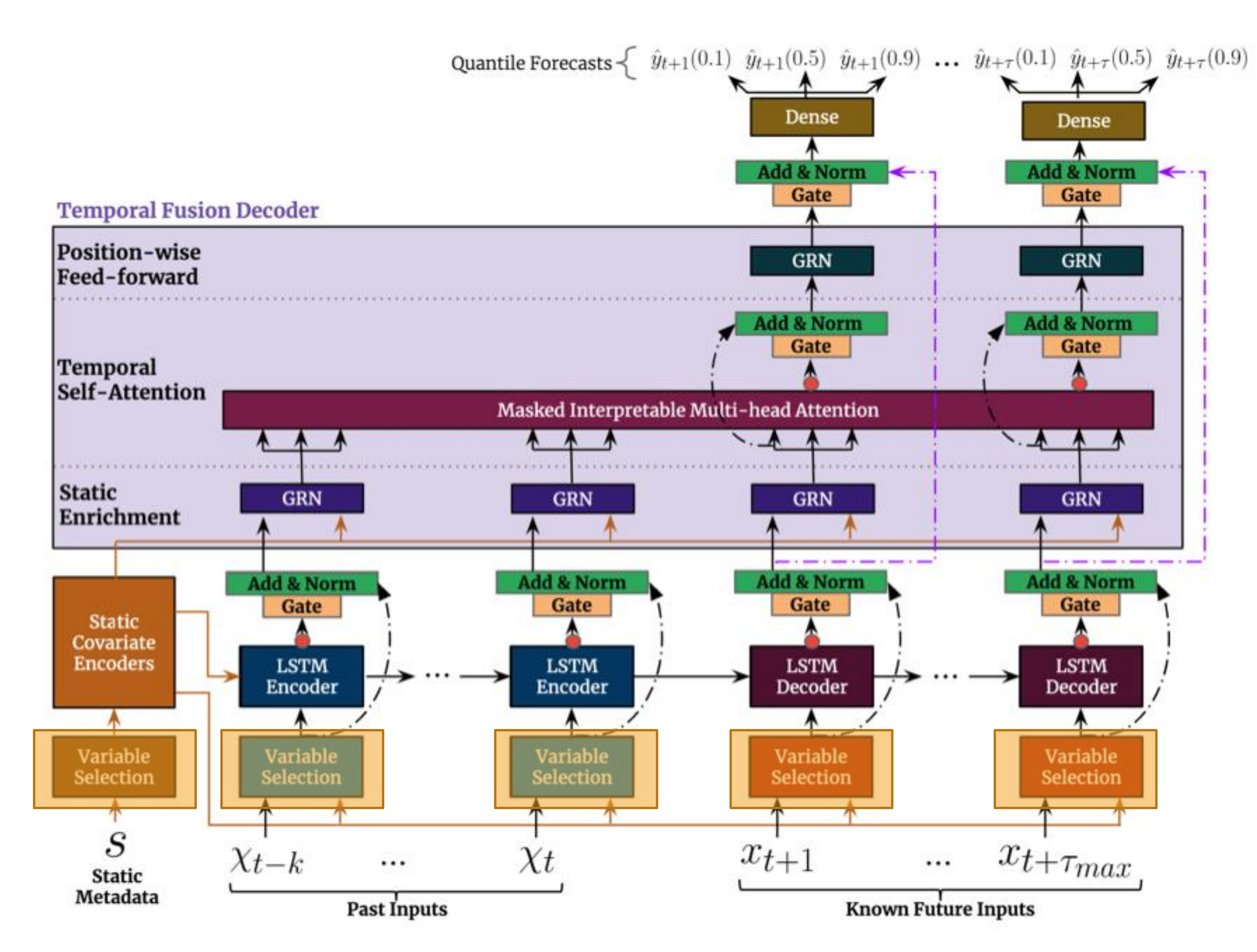
(5)-1 가 value scaling 역할

Variable Selection Networks = VSN

각 시점 input 의 여러 features 중
예측값에 확실히 관여하는 알맹이들만 남기기

VSN layer 는 GRN layer 을 포함
모든 inputs 는 VSN layer 을 거침





Variable Selection Networks = VSN

12월 아이스크림의 예측 판매량은 ?

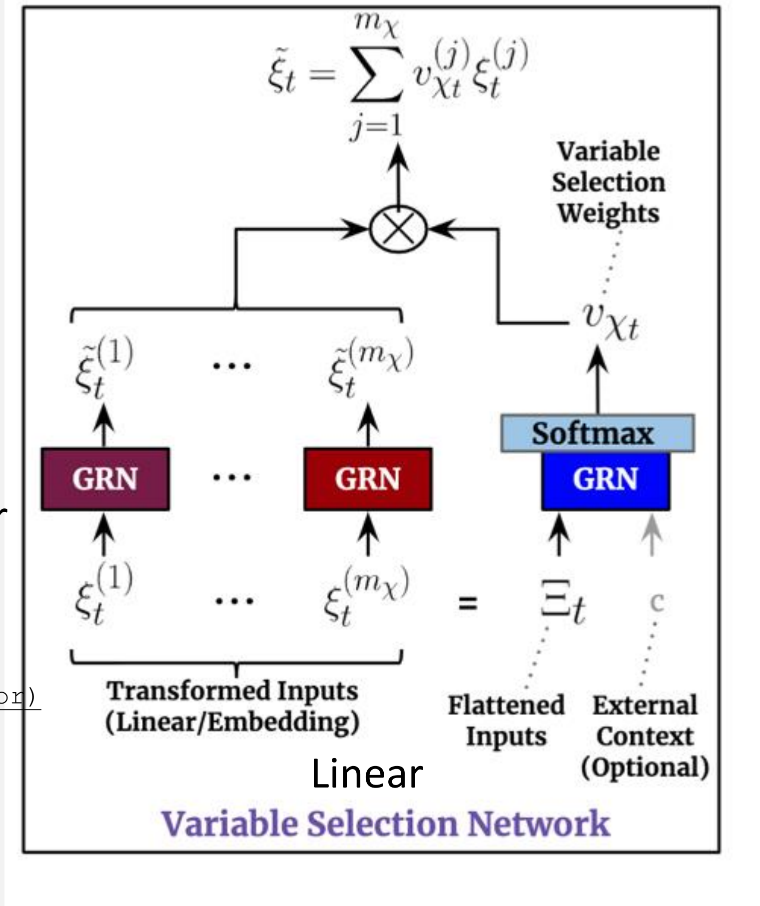
T 일 input 값

공휴일 여부	엄마는 외계인	민트초코
o	200개	100개

Categorical

O → Entity embedding (D Model vector)

Non-Linear



Continuous

Linear transformation nn.linear(1,D Model vector)

200 →

100 →

Variable Selection Weights

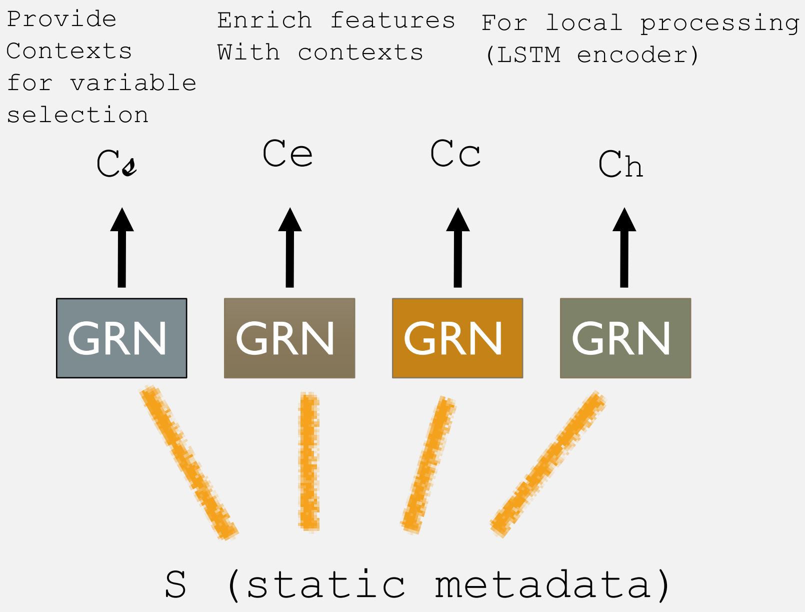
Flattened Inputs

Feature 개수만큼 (j개)

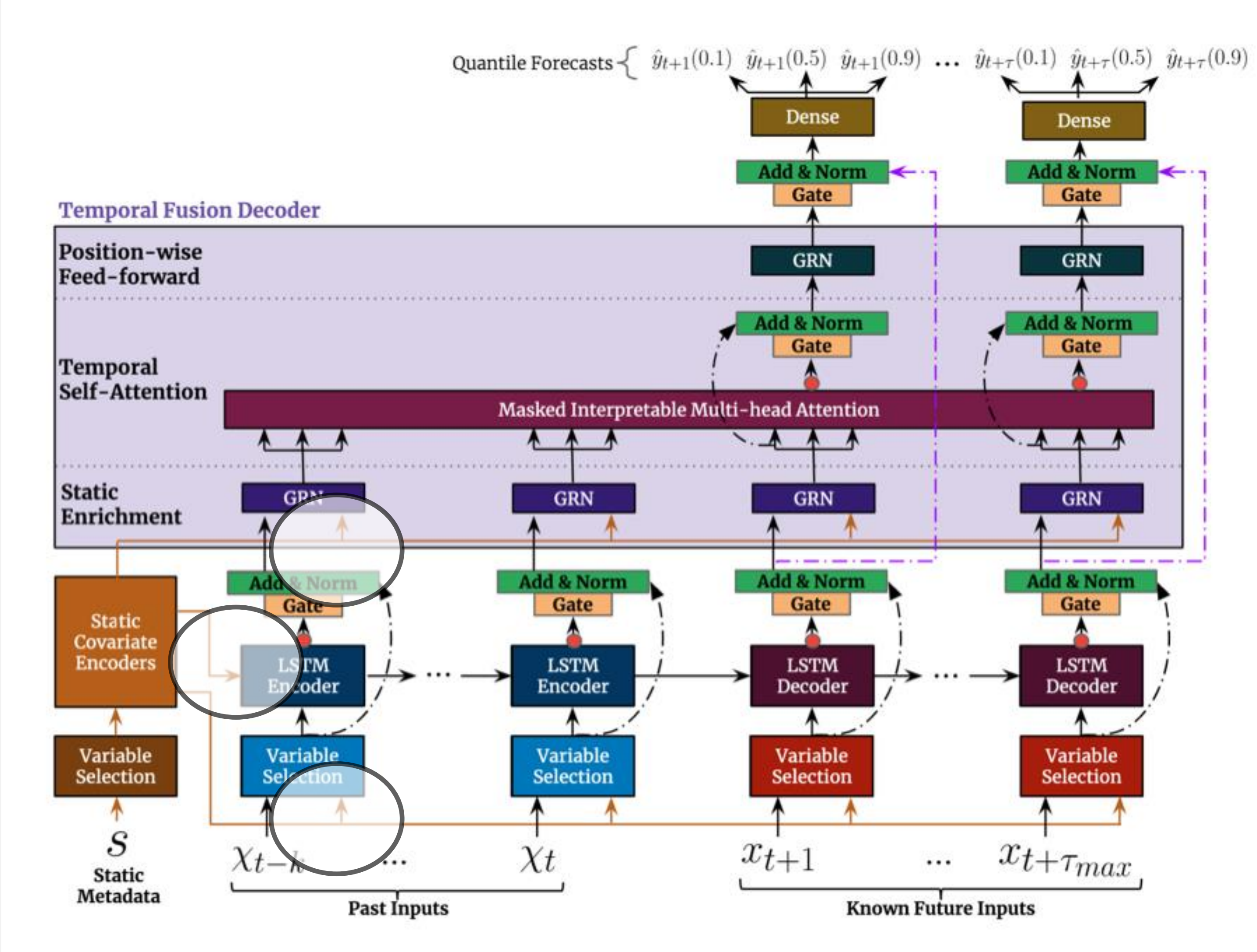
0.7	0.2	0.1

Static Covariate Encoders

S 메타 데이터를 features 을 이해할 수 있는 context 로 사용



각기 다른 4개의 GRN 을 사용하여서
쓰임이 다른 4개의 문맥 생성

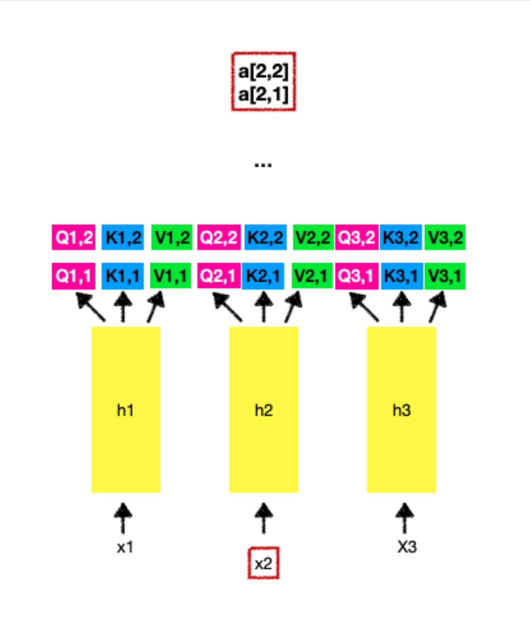


Interpretable Multi-Head Attention

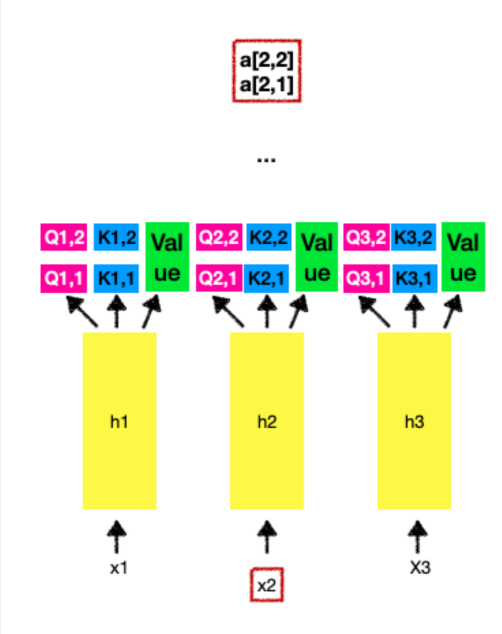
각 time step 의 장기간 상호관계 도출

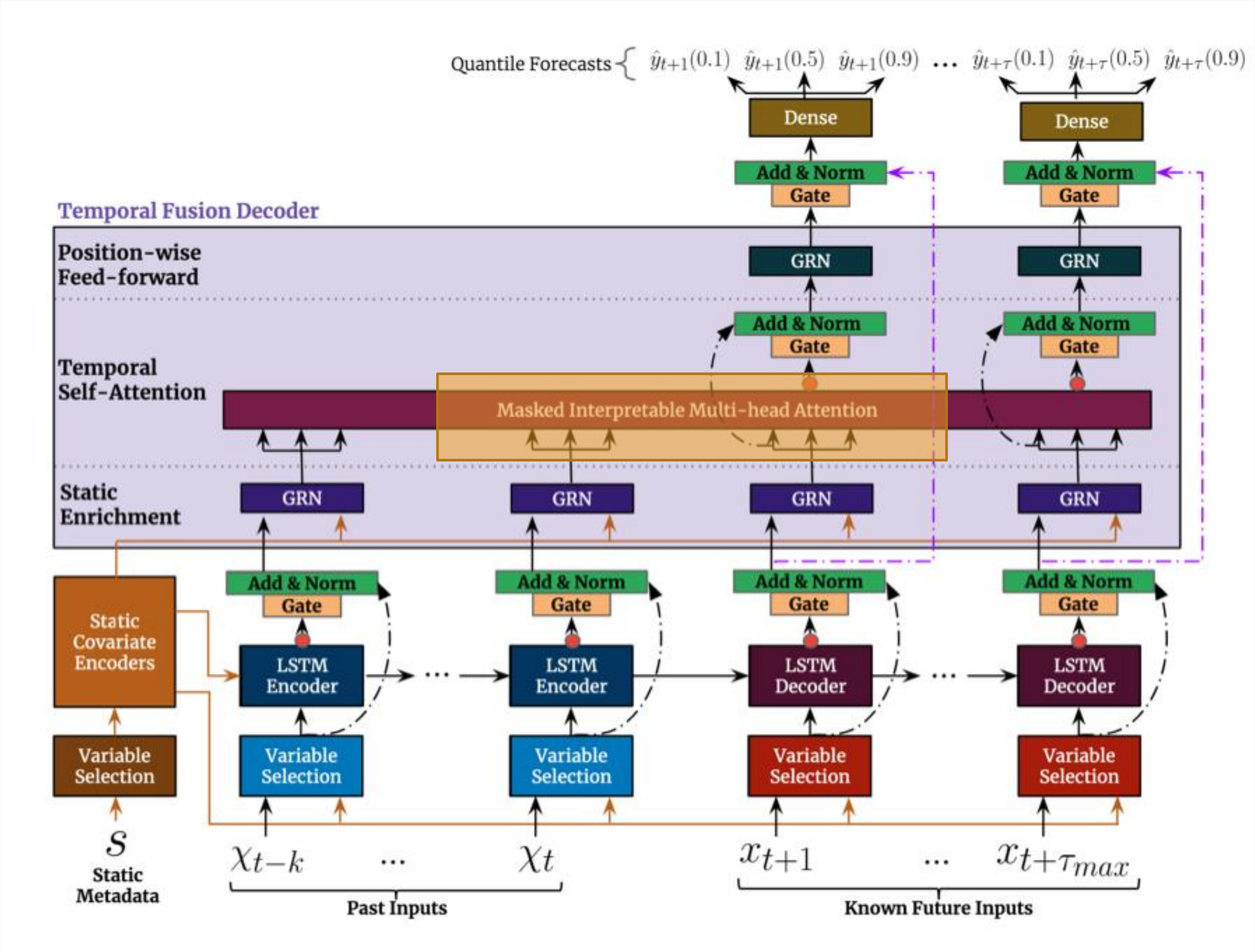
Multi-attention 아키텍처 그대로 갖고가되,
query,key,value 중 value 는 모든 head 에서 동일

원 Multi-head attention



TFT Multi-head attention





Interpretable Multi-Head Attention

각 time step 의 장기간 상호관계 도출

같은 timestep 은 다른 head 에서도
동일한 value 를 갖게 함으로써 앙상블하는 방식으로 작용

TFT Multi-head attention

$$\mathrm{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = [\boldsymbol{H}_1,\dots,\boldsymbol{H}_{m_H}]\,\boldsymbol{W}_H, \tag{11}$$

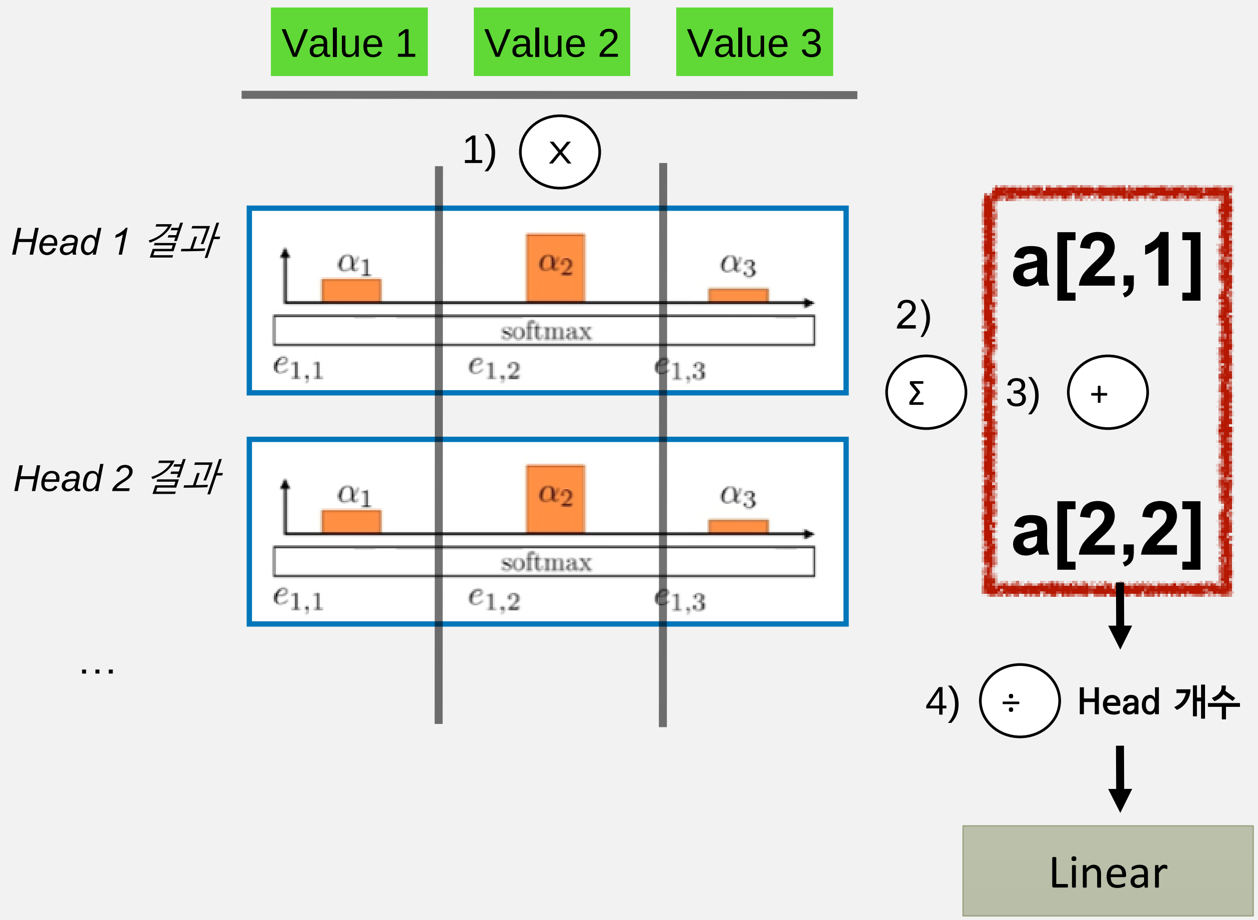
$$\boldsymbol{H}_h = \mathrm{Attention}(\boldsymbol{Q}\,\boldsymbol{W}_Q^{(h)},\boldsymbol{K}\,\boldsymbol{W}_K^{(h)},\boldsymbol{V}\,\boldsymbol{W}_V^{(h)}), \tag{12}$$

$$\mathrm{InterpretableMultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \tilde{\boldsymbol{H}}\,\boldsymbol{W}_H, \tag{13}$$

$$\tilde{\boldsymbol{H}} = \tilde{A}(\boldsymbol{Q},\boldsymbol{K})\,\boldsymbol{V}\,\boldsymbol{W}_V, \tag{14}$$

$$= \left\{1/H\sum_{h=1}^{m_H} A\left(\boldsymbol{Q}\,\boldsymbol{W}_Q^{(h)},\boldsymbol{K}\,\boldsymbol{W}_K^{(h)}\right)\right\}\boldsymbol{V}\,\boldsymbol{W}_V, \tag{15}$$

$$= 1/H\sum_{h=1}^{m_H} \mathrm{Attention}(\boldsymbol{Q}\,\boldsymbol{W}_Q^{(h)},\boldsymbol{K}\,\boldsymbol{W}_K^{(h)},\boldsymbol{V}\,\boldsymbol{W}_V), \tag{16}$$



ex. Timestep 2 의 어텐션 결과

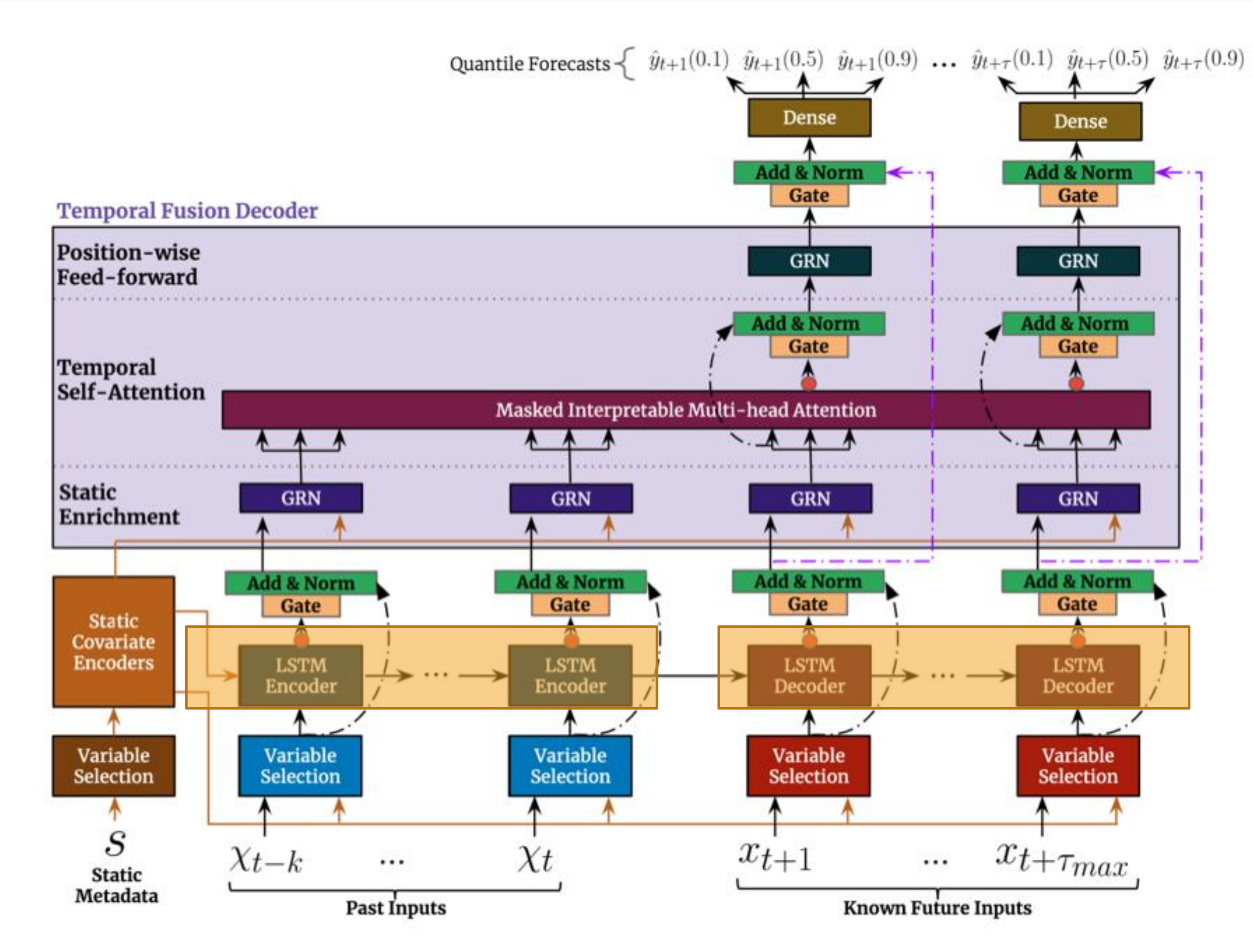
=> 앙상블 느낌 !

Temporal Fusion Decoder - (1)
Seq2Seq layer

Temporal Fusion Decoder 에 들어가는 final inputs

각 시점의 특징 추출, 각 시점 정보 추가

$$\tilde{\phi}(t,n) = \text{LayerNorm}\left(\tilde{\xi}_{t+n} + \text{GLU}_{\tilde{\phi}}(\phi(t,n))\right), \quad (17)$$

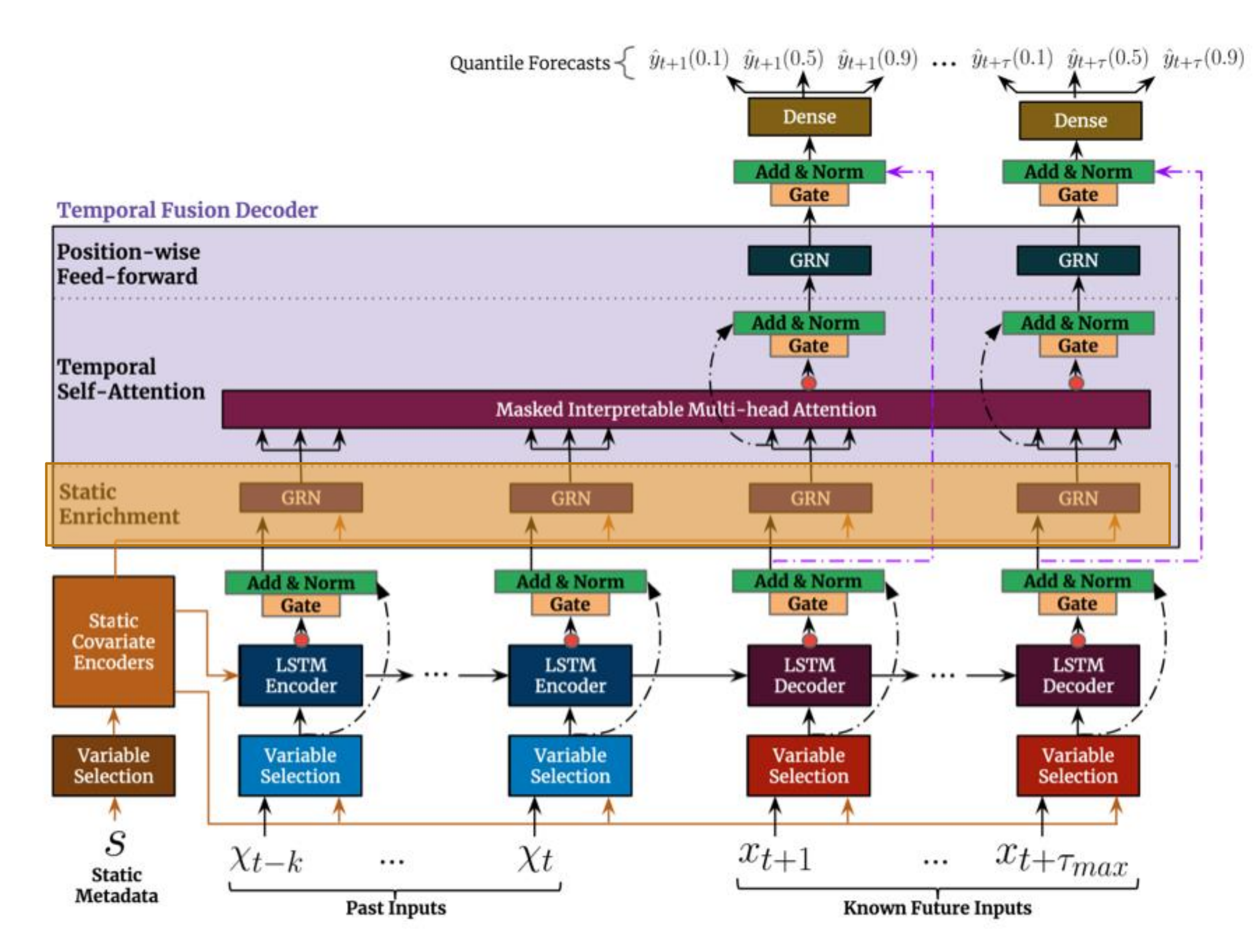


Temporal Fusion Decoder - (2)
Static Enrichment Layer

Temporal Fusion Decoder 에 들어가는 final inputs

temporal features 에 메타데이터를 활용해 풍부한 문맥 추가

$$\boldsymbol{\theta}(t,n) = \mathrm{GRN}_{\theta}\left(\tilde{\boldsymbol{\phi}}(t,n), \boldsymbol{c}_e\right), \tag{18}$$



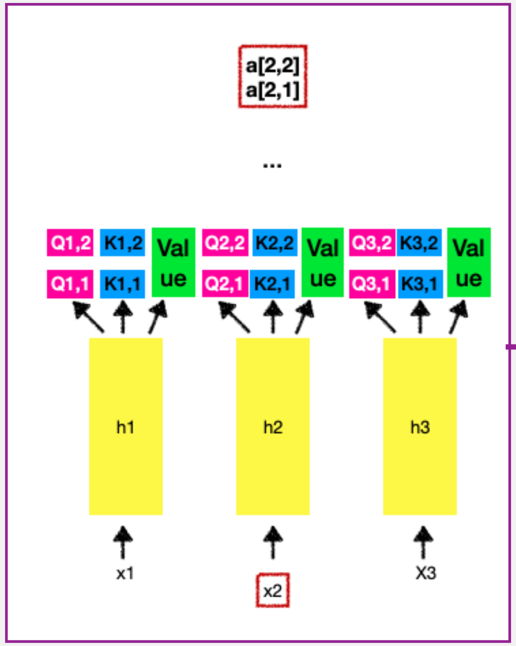
Temporal Fusion Decoder - (3)
Temporal Self-Attention Layer

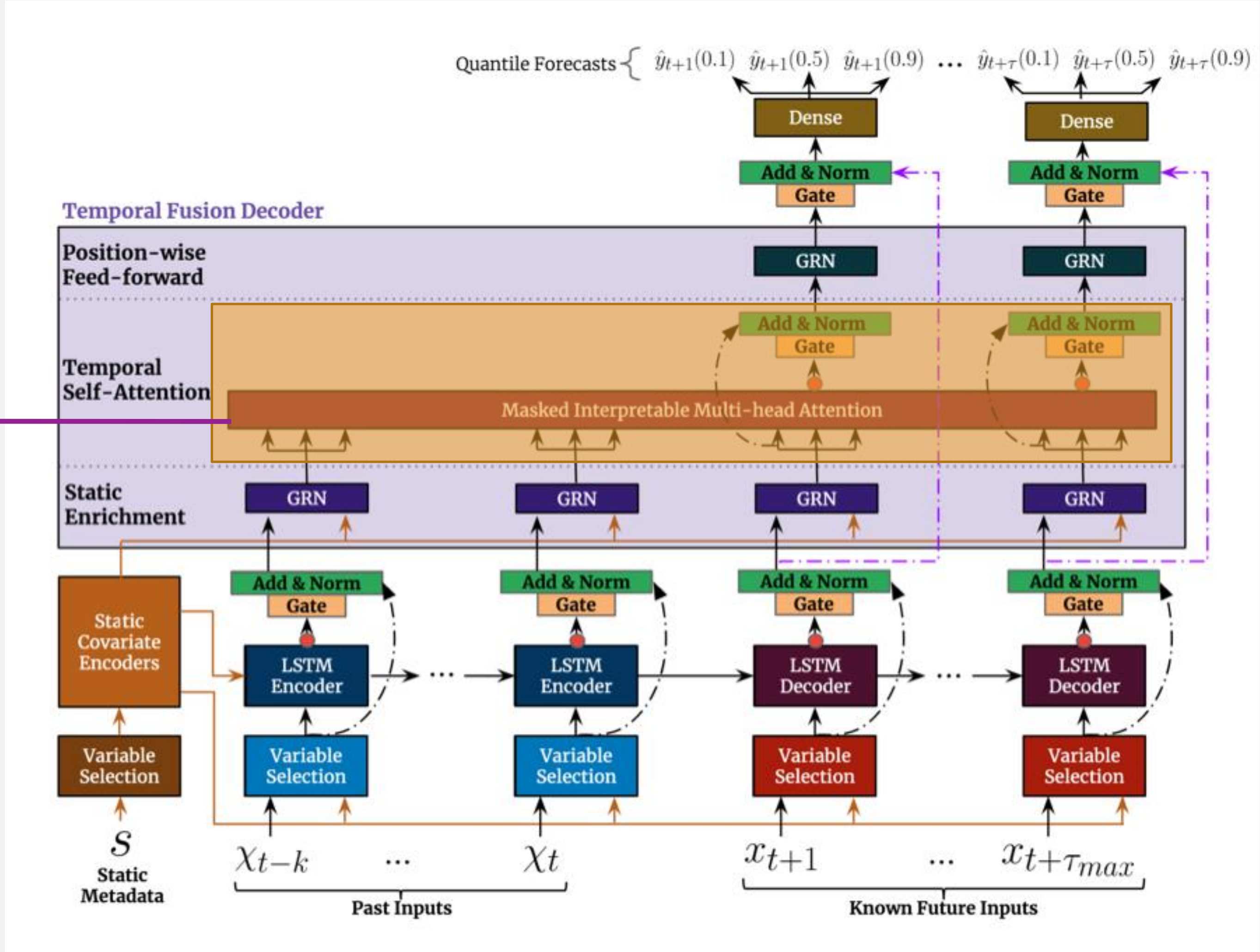
Interpretable Multi-Head Attention

이 투입된 layer

각 time step 의 장기간 상호관계 도출

TFT `Multi-head attention`





Temporal Fusion Decoder - (3)
Temporal Self-Attention Layer

Interpretable Multi-Head Attention 이 투입된 layer

각 time step 의 장기간 상호관계 도출

$$\theta(t) = [\theta(t,-k), \ldots, \theta(t,\tau)]^T$$

Static enrichment layer 에서 나온 값들을 하나의 단일 벡터로 뭉치기

↓

TFT masked Multi-head attention : 이전 시점들과의 관계(;attention) 만을 이용하도록 하기 위해서

↓

$$\delta(t,n) = \text{LayerNorm}(\boldsymbol{\theta}(t,n) + \text{GLU}_\delta(\boldsymbol{\beta}(t,n))). \qquad (20)$$

n > t

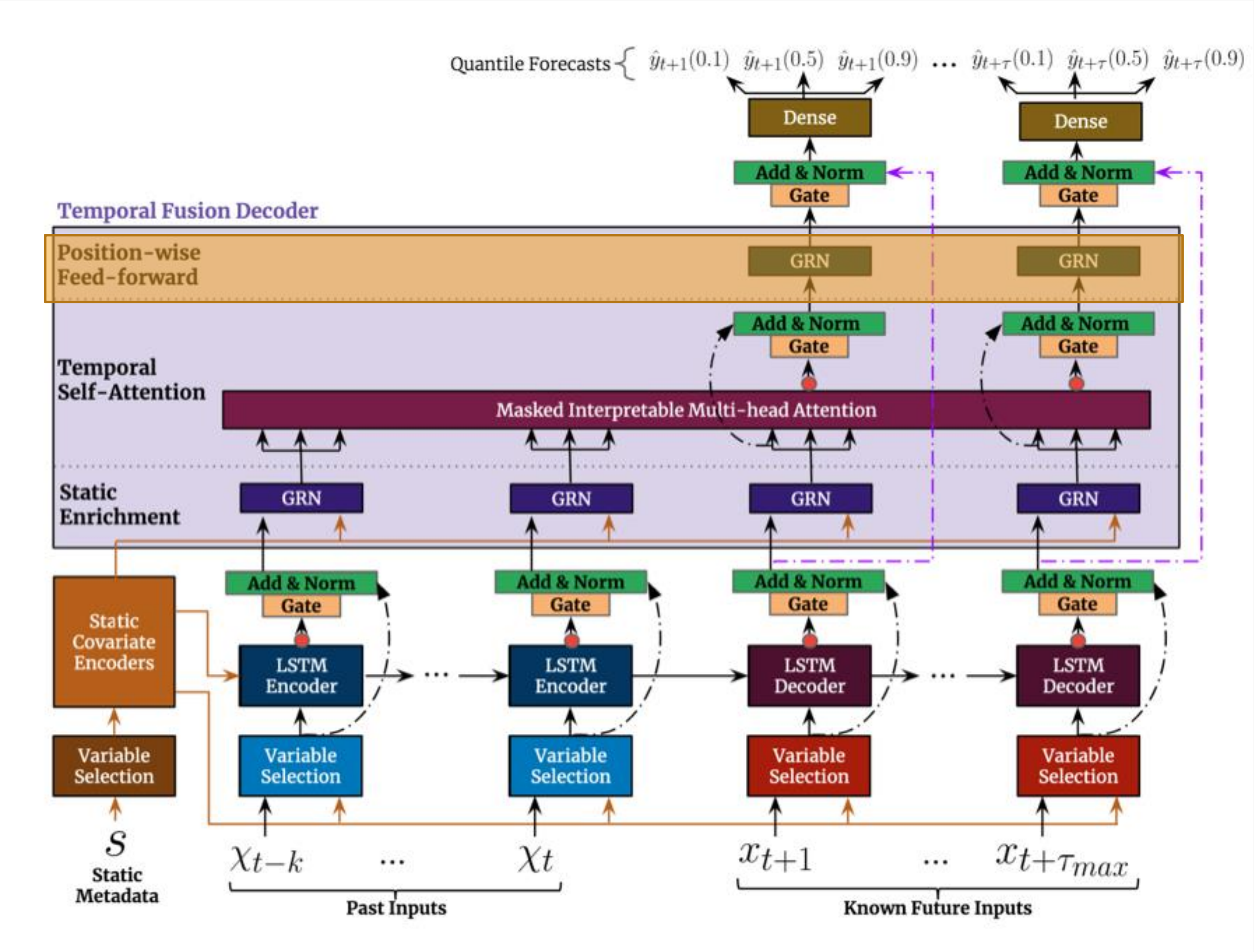
Temporal Fusion Decoder - (4)
Position-wise Feed-forward layer

n > t

non-linear 층 (GRN) 추가

$$\psi(t,n) = \mathrm{GRN}_{\psi}\left(\boldsymbol{\delta}(t,n)\right), \tag{21}$$

$$\tilde{\psi}(t,n) = \mathrm{LayerNorm}\left(\tilde{\phi}(t,n) + \mathrm{GLU}_{\tilde{\psi}}(\psi(t,n))\right), \tag{22}$$

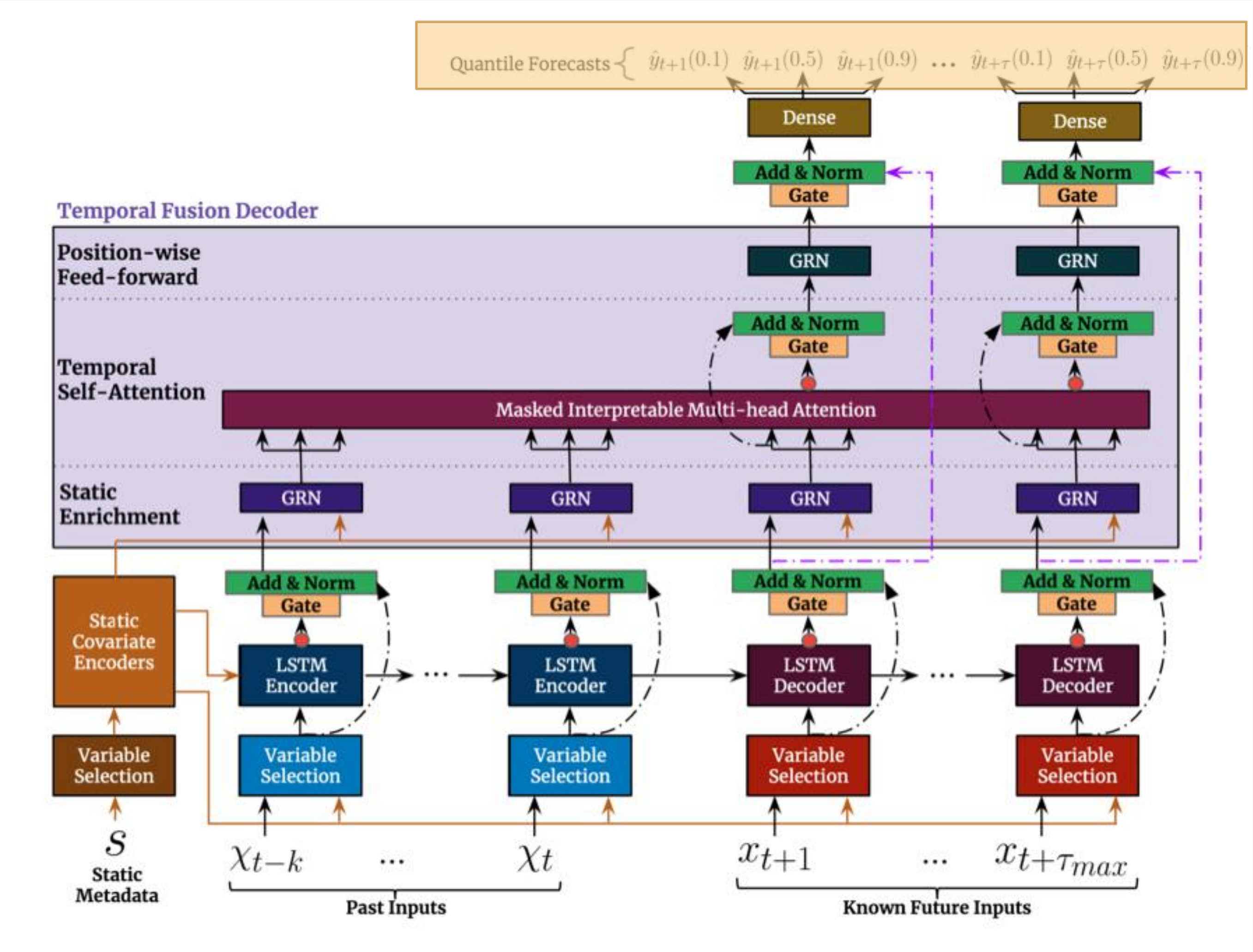


Quantile Outputs

각 quantile 마다 각자의 linear 층으로 output 값 계산
Quantile : 해당 예측값이 나올 확률이 quantile %

최종 quantile 확률별 outputs 출력

$$\hat{y}(q, t, \tau) = W_q \tilde{\psi}(t, \tau) + b_q, \qquad (23)$$



4. LOSS FUNCTION

LOSS FUNCTION

해당 데이터셋 内

$$\mathcal{L}(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1}^{\tau_{max}} \frac{QL\left(y_t,\ \hat{y}(q, t-\tau, \tau),\ q\right)}{M\tau_{max}} \quad (24)$$

{0.1,0.5,0.9}

예측할 개수

$$QL(y,\ \hat{y},\ q) = q(y-\hat{y})_{+} + (1-q)(\hat{y}-y)_{+}, \quad (25)$$

max(0,x)

- Quantile loss function 출처 : https://arxiv.org/pdf/1711.11053.pdf
- 조금 변형된 loss function 으로 out of sample test 도 같이 진행

$$q\text{-Risk} = \frac{2\sum_{y_t \in \tilde{\Omega}} \sum_{\tau=1}^{\tau_{max}} QL\left(y_t,\ \hat{y}(q, t-\tau, \tau),\ q\right)}{\sum_{y_t \in \tilde{\Omega}} \sum_{\tau=1}^{\tau_{max}} |y_t|}, \quad (26)$$

where $\tilde{\Omega}$ is the domain of test samples. Full details on hyperparameter optimization and training can be found in Appendix A.

5. 데이터 셋 / 실험 결과

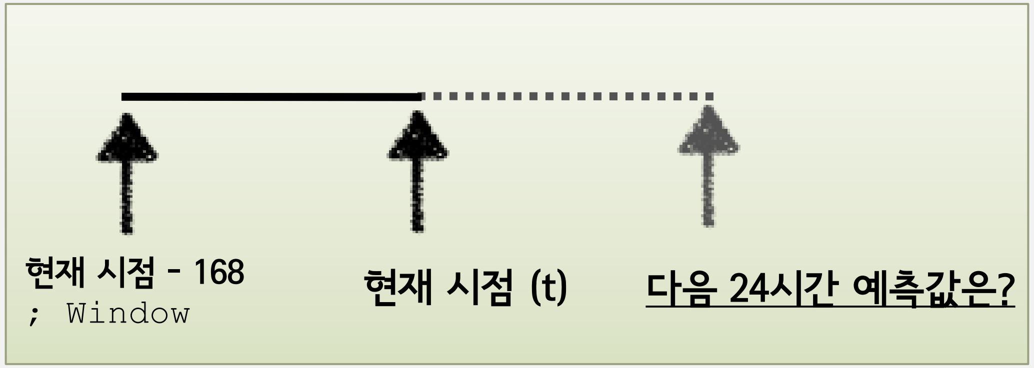
데이터 셋 1 – The UCI electricity Load Diagrams Dataset

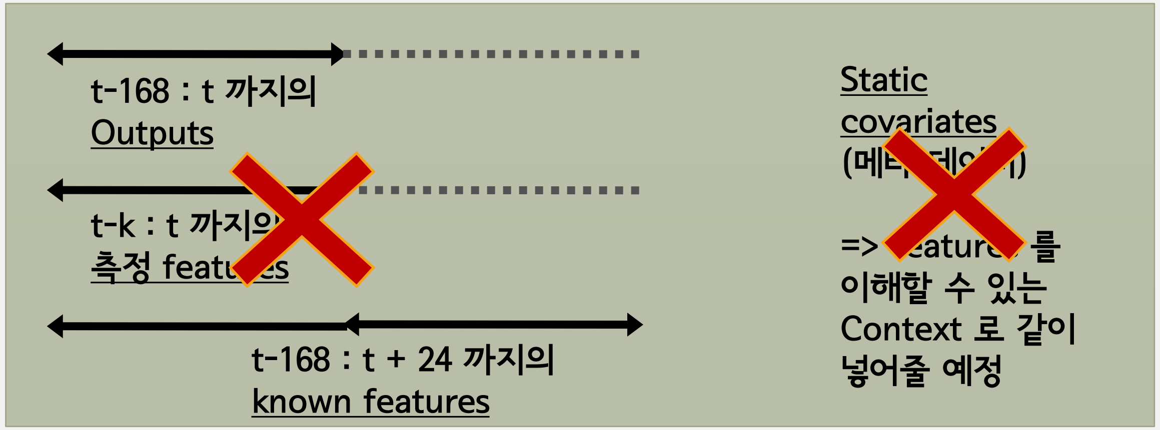
Electricity consumption of 370 customers (168 x 370)

총 370 명

Id	Hours_from_start	Power_usage	hour	Day_of_week	Hours_from_start	Categorical_id
ID	TIME	TARGET	KNOWN_INPUT	KNOWN_INPUT	KNOWN_INPUT	STATIC_INPUT
REAL_VALUED	REAL_VALUED	REAL_VALUED	REAL_VALUED	REAL_VALUED	REAL_VALUED	CATEGORICAL

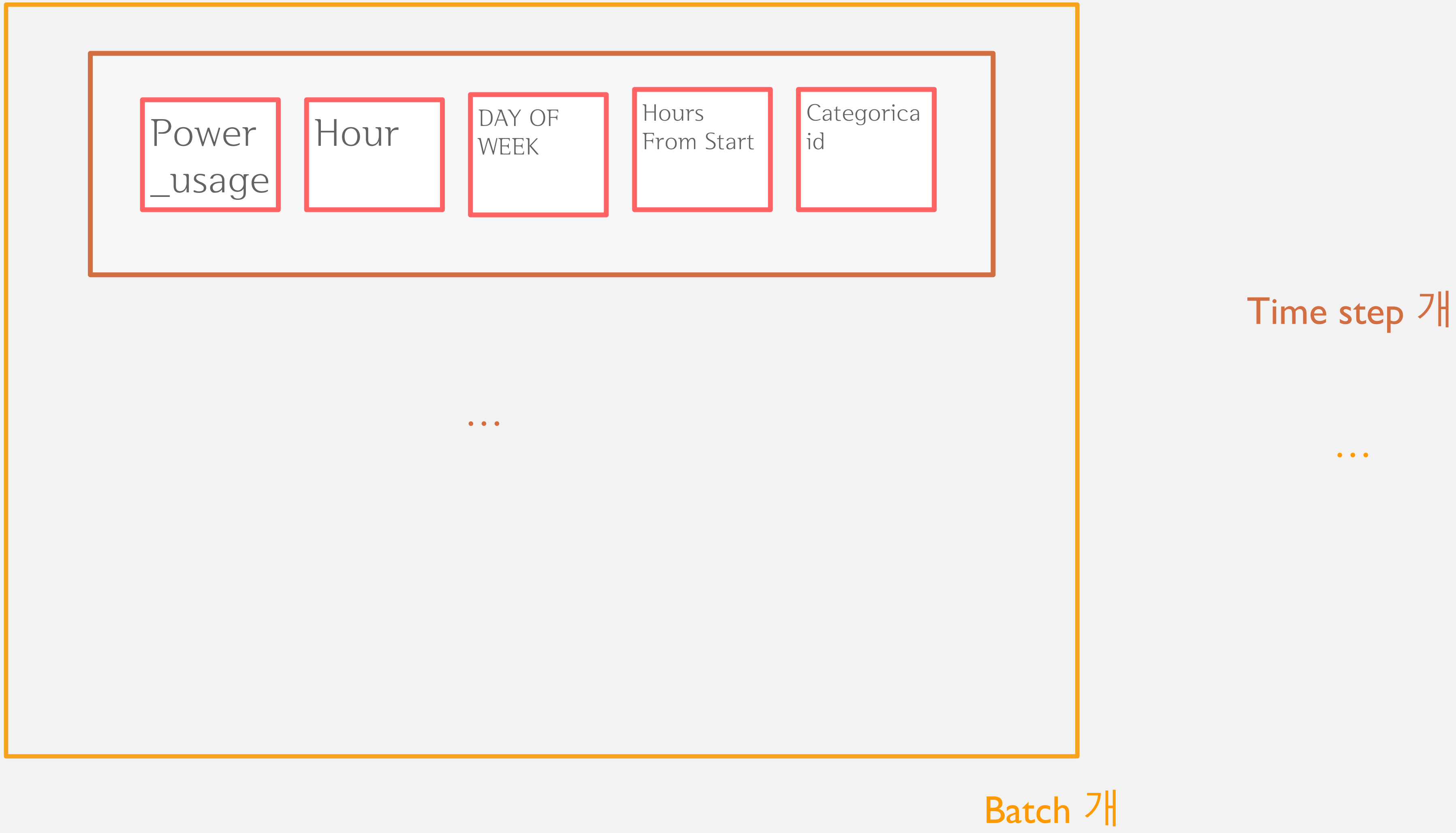
이용할 features ~~4가지~~ 2가지





per id (=itendity)

```
def _batch_data(self,data):
    .......
# Returns : Batched Numpy array with shape ( ?,self.time_steps,self.input_size)
```



데이터 셋 1 – The UCI electricity Load Diagrams Dataset

Electricity consumption of 370 customers

1 ~ 168 / 다음 24시간을 예측 (169 ~192)

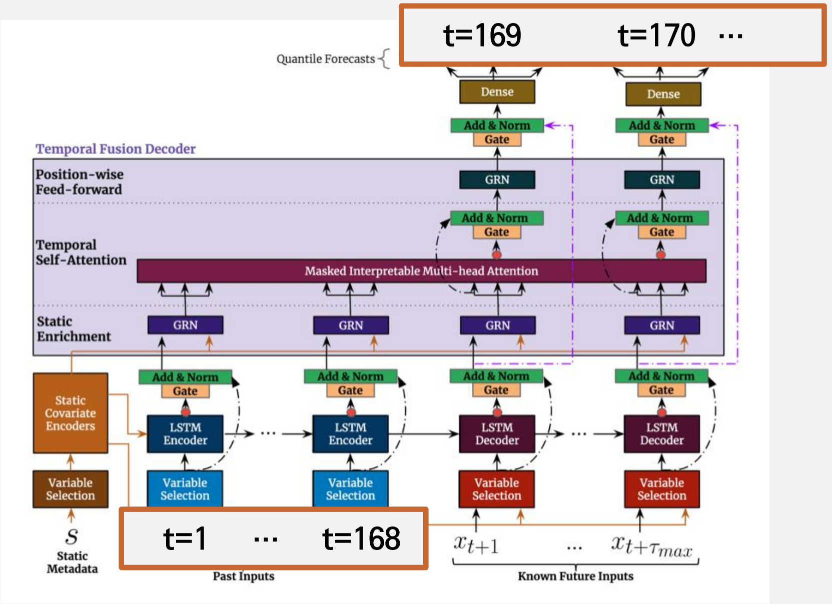
t=192

$\mathcal{T}$=24

$$\mathcal{L}(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1} \frac{QL\left(y_t,\ \hat{y}(q, t-\tau, \tau),\ q\right)}{M \tau_{max}} \qquad (24)$$

t=169

$$QL(y,\ \hat{y},\ q) = q(y - \hat{y})_+ + (1-q)(\hat{y} - y)_+, \qquad (25)$$



2 ~ 169 / 다음 24시간을 예측 (193 ~384)

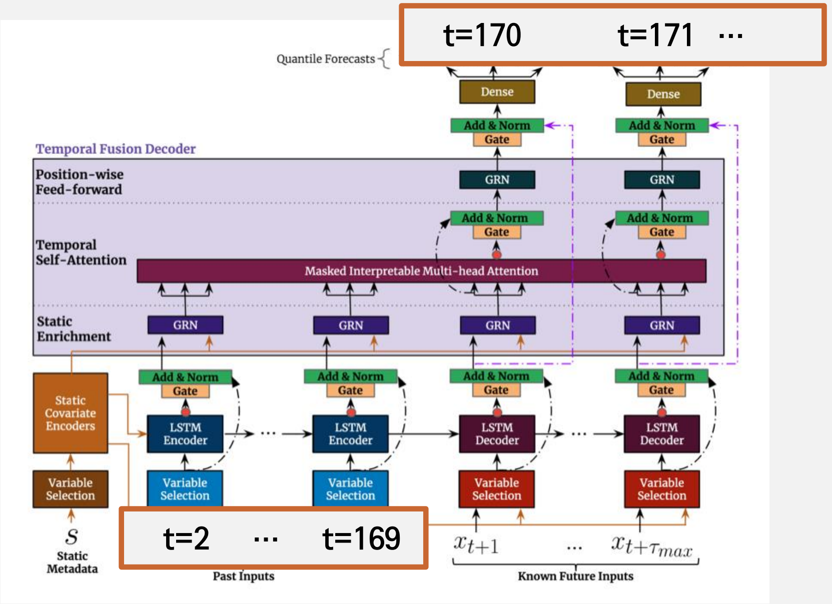
t=384

$\mathcal{T}$=24

$$\mathcal{L}(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1} \frac{QL\left(y_t,\ \hat{y}(q, t-\tau, \tau),\ q\right)}{M \tau_{max}} \qquad (24)$$

t=193

$$QL(y,\ \hat{y},\ q) = q(y - \hat{y})_+ + (1-q)(\hat{y} - y)_+, \qquad (25)$$

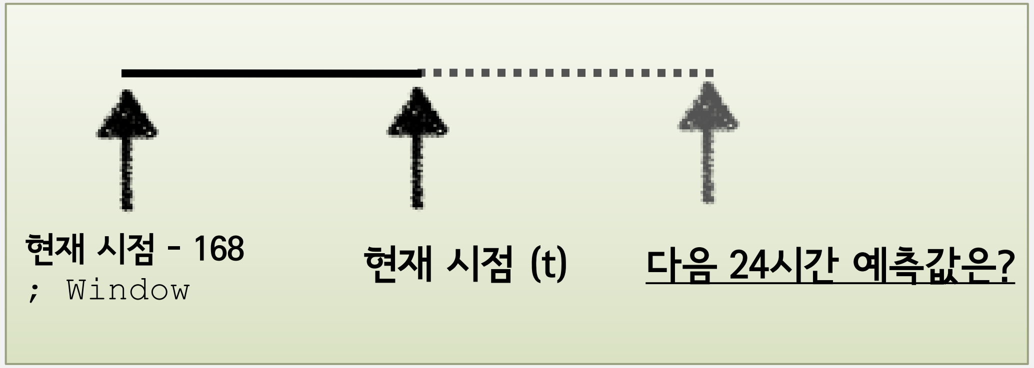


...

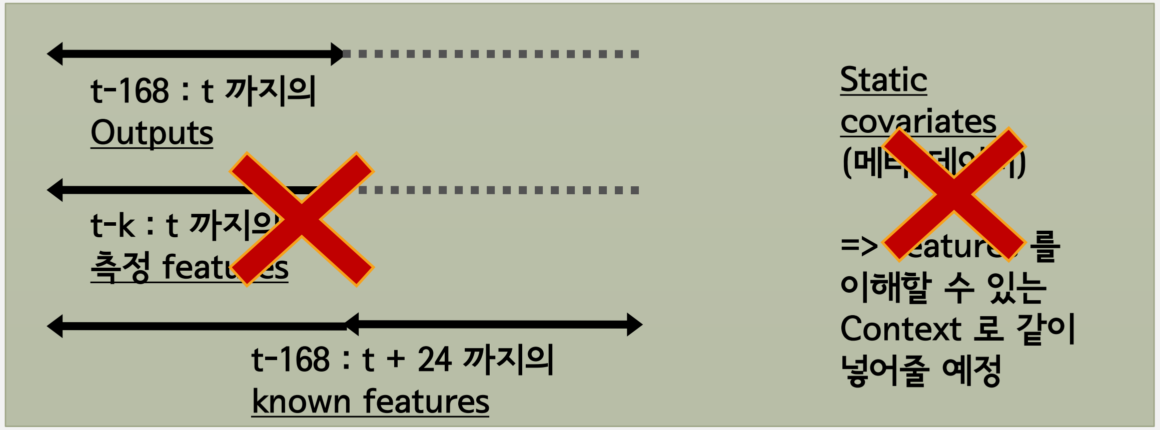
데이터 셋 2 – The UCI PEM-SF Traffic Dataset

총 440 route freeways

Id	Hours_from_start	values	Time_on_day	Day_of_week	Hours_from_start	Categorical_id
ID	TIME	TARGET	KNOWN_INPUT	KNOWN_INPUT	KNOWN_INPUT	STATIC_INPUT
REAL_VALUED	REAL_VALUED	REAL_VALUED	REAL_VALUED	REAL_VALUED	REAL_VALUED	CATEGORICAL



이용할 features ~~4가지~~ 2가지



데이터 셋 3 – Favorita Grocery Sales Dataset

메타데이터 4개

총 8개의 dataset https://www.kaggle.com/c/favorita-grocery-sales-forecasting/rules

items

item_nbr	family	class	perishable
96995	GROCERY I	1093	0
99197	GROCERY I	1067	0
103501	CLEANING	3008	0
103520	GROCERY I	1028	0
103665	BREAD/BAKERY	2712	1
105574	GROCERY I	1045	0
105575	GROCERY I	1045	0
105576	GROCERY I	1045	0
105577	GROCERY I	1045	0
105693	GROCERY I	1034	0

stores

store_nbr	city	state	type	cluster
1	Quito	Pichincha	D	13
2	Quito	Pichincha	D	13
3	Quito	Pichincha	D	8
4	Quito	Pichincha	D	9
5	Santo Domingo	Santo Domingo de los Tsachilas	D	4
6	Quito	Pichincha	D	13
7	Quito	Pichincha	D	8
8	Quito	Pichincha	D	8
9	Quito	Pichincha	B	6
10	Quito	Pichincha	C	15

oil

date	dcoilwtico
2013-01-01	
2013-01-02	93.14
2013-01-03	92.97
2013-01-04	93.12
2013-01-07	93.2
2013-01-08	93.21
2013-01-09	93.08
2013-01-10	93.81
2013-01-11	93.6
2013-01-14	94.27
2013-01-15	93.26

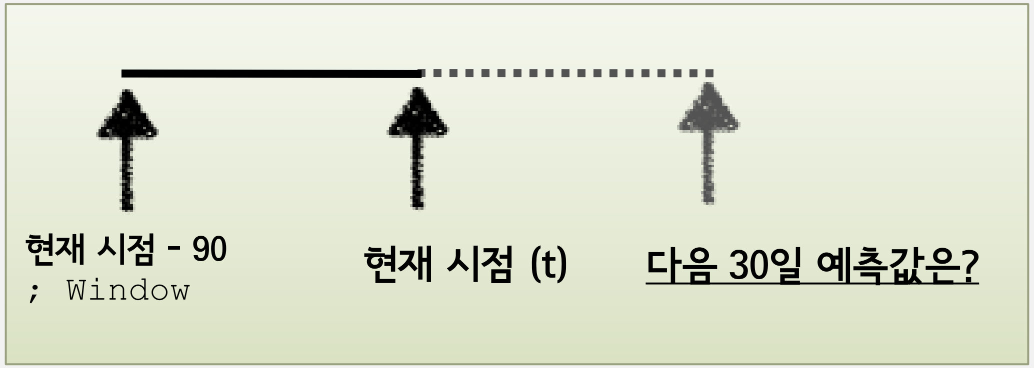
holidays_events

date	type	locale	locale_name	description	transferred
2012-03-02	Holiday	Local	Manta	Fundacion de Manta	FALSE
2012-04-01	Holiday	Regional	Cotopaxi	Provincializacion de Cotopaxi	FALSE
2012-04-12	Holiday	Local	Cuenca	Fundacion de Cuenca	FALSE
2012-04-14	Holiday	Local	Libertad	Cantonizacion de Libertad	FALSE
2012-04-21	Holiday	Local	Riobamba	Cantonizacion de Riobamba	FALSE
2012-05-12	Holiday	Local	Puyo	Cantonizacion del Puyo	FALSE
2012-06-23	Holiday	Local	Guaranda	Cantonizacion de Guaranda	FALSE
2012-06-25	Holiday	Regional	Imbabura	Provincializacion de Imbabura	FALSE
2012-06-25	Holiday	Local	Latacunga	Cantonizacion de Latacunga	FALSE

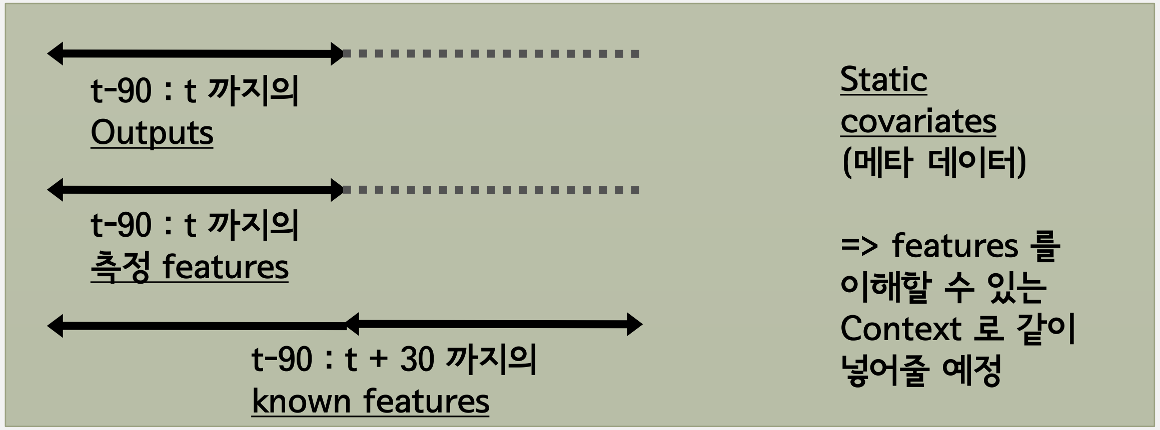
데이터 셋 3 – Favorita Grocery Sales Dataset

총 130k

Traj_id	date	Log_sales	Onpromotion	transactions	oil	Day_of_week	Day_of_month	month	National_hol	Regional_hol	Local_hol	open	Item_nbr	Store_nbr	city	state	type	cluster	family	class	perishable
ID	TIME	TARGET	KNOWN	OBSERVED	OBSERVED	KNOWN	KNOWN	KNOWN	KNOWN	KNOWN	KNWON	KNOWN	STATIC	STATIC	STATIC	STATIC	STATIC	STATIC	STATIC	STATIC	STATIC
REAL	DATE	REAL	CATEGORICAL	REAL	REAL	CATEGORICAL	REAL	REAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	REAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL



이용할 features 4가지

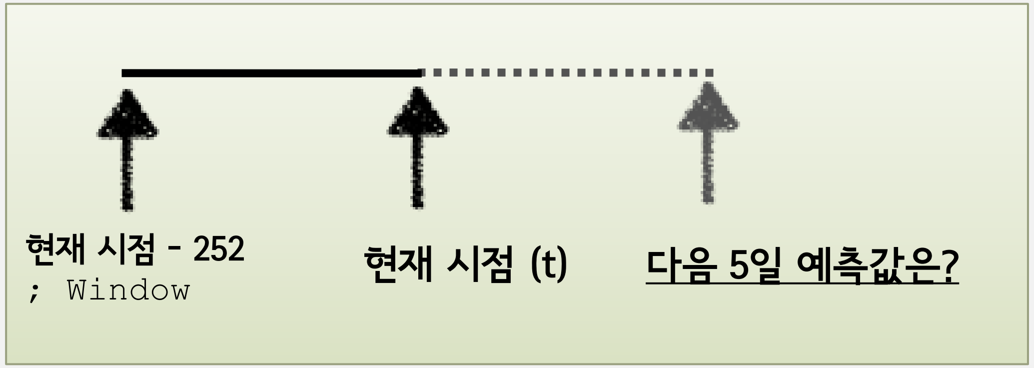


데이터 셋 4 – The OMI realized library

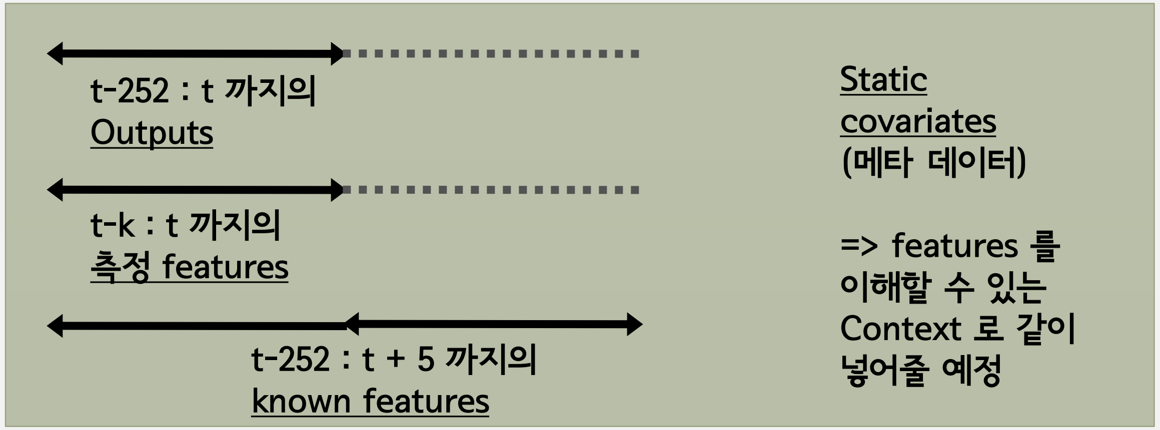
총 41

Symbol	date	Log_vol	Open_to_close	Days_from_start	Day_of_week	Day_of_month	Week_of_year	month	region
ID	TIME	TARGET	OBSERVED	KNOWN	KNOWN	KNOWN	KNOWN	KNOWN	STATIC
CATEGORICAL	DATE	REAL	REAL	REAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL	CATEGORICAL

...



이용할 features ~~4가지~~ 2가지



실험 결과

- 하이퍼파라미터 정보 (random search 를 해가면서 발견 – validation loss 기준)

Table 1: Information on dataset and optimal TFT configuration.

	Electricity	Traffic	Retail	Vol.
Dataset Details				
Target Type	$\mathbb{R}$	[0, 1]	$\mathbb{R}$	$\mathbb{R}$
Number of Entities	370	440	130k	41
Number of Samples	500k	500k	500k	∼100k
Network Parameters				
k	168	168	90	252
τ_{max}	24	24	30	5
Dropout Rate	0.1	0.3	0.1	0.3
State Size	160	320	240	160
Number of Heads	4	4	4	1
Training Parameters				
Minibatch Size	64	128	128	64
Learning Rate	0.001	0.001	0.001	0.01
Max Gradient Norm	0.01	100	100	0.01

GRN 을 이용하여 효과적으로 연산 비용을 줄임.
(V100) train – 6시간, validate – 8분 소요

실험 결과

Table 2: P50 and P90 quantile losses on a range of real-world datasets. Percentages in brackets reflect the increase in quantile loss versus TFT (lower q-Risk better), with TFT outperforming competing methods across all experiments, improving on the next best alternative method (underlined) between 3% and 26%.

	ARIMA	ETS	TRMF	DeepAR	DSSM
Electricity	0.154 (*+180%*)	0.102 (*+85%*)	0.084 (*+53%*)	0.075 (*+36%*)	0.083 (*+51%*)
Traffic	0.223 (*+135%*)	0.236 (*+148%*)	0.186 (*+96%*)	0.161 (*+69%*)	0.167 (*+76%*)
	ConvTrans	**Seq2Seq**	**MQRNN**	**TFT**	
Electricity	0.059 (*+7%*)	0.067 (*+22%*)	0.077 (*+40%*)	**0.055***	
Traffic	0.122 (*+28%*)	0.105 (*+11%*)	0.117 (*+23%*)	**0.095***	

(a) P50 losses on simpler univariate datasets.

	ARIMA	ETS	TRMF	DeepAR	DSSM
Electricity	0.102 (*+278%*)	0.077 (*+185%*)	-	0.040 (*+48%*)	0.056 (*+107%*)
Traffic	0.137 (*+94%*)	0.148 (*+110%*)	-	0.099 (*+40%*)	0.113 (*+60%*)
	ConvTrans	**Seq2Seq**	**MQRNN**	**TFT**	
Electricity	0.034 (*+26%*)	0.036 (*+33%*)	0.036 (*+33%*)	**0.027***	
Traffic	0.081 (*+15%*)	0.075 (*+6%*)	0.082 (*+16%*)	**0.070***	

(b) P90 losses on simpler univariate datasets.

	DeepAR	CovTrans	Seq2Seq	MQRNN	TFT
Vol.	0.050 (*+28%*)	0.047 (*+20%*)	0.042 (*+7%*)	0.042 (*+7%*)	**0.039***
Retail	0.574 (*+62%*)	0.429 (*+21%*)	0.411 (*+16%*)	0.379 (*+7%*)	**0.354***

(c) P50 losses on datasets with rich static or observed inputs.

	DeepAR	CovTrans	Seq2Seq	MQRNN	TFT
Vol.	0.024 (*+21%*)	0.024 (*+22%*)	0.021 (*+8%*)	0.021 (*+9%*)	**0.020***
Retail	0.230 (*+56%*)	0.192 (*+30%*)	0.157 (*+7%*)	0.152 (*+3%*)	**0.147***

(d) P90 losses on datasets with rich static or observed inputs.

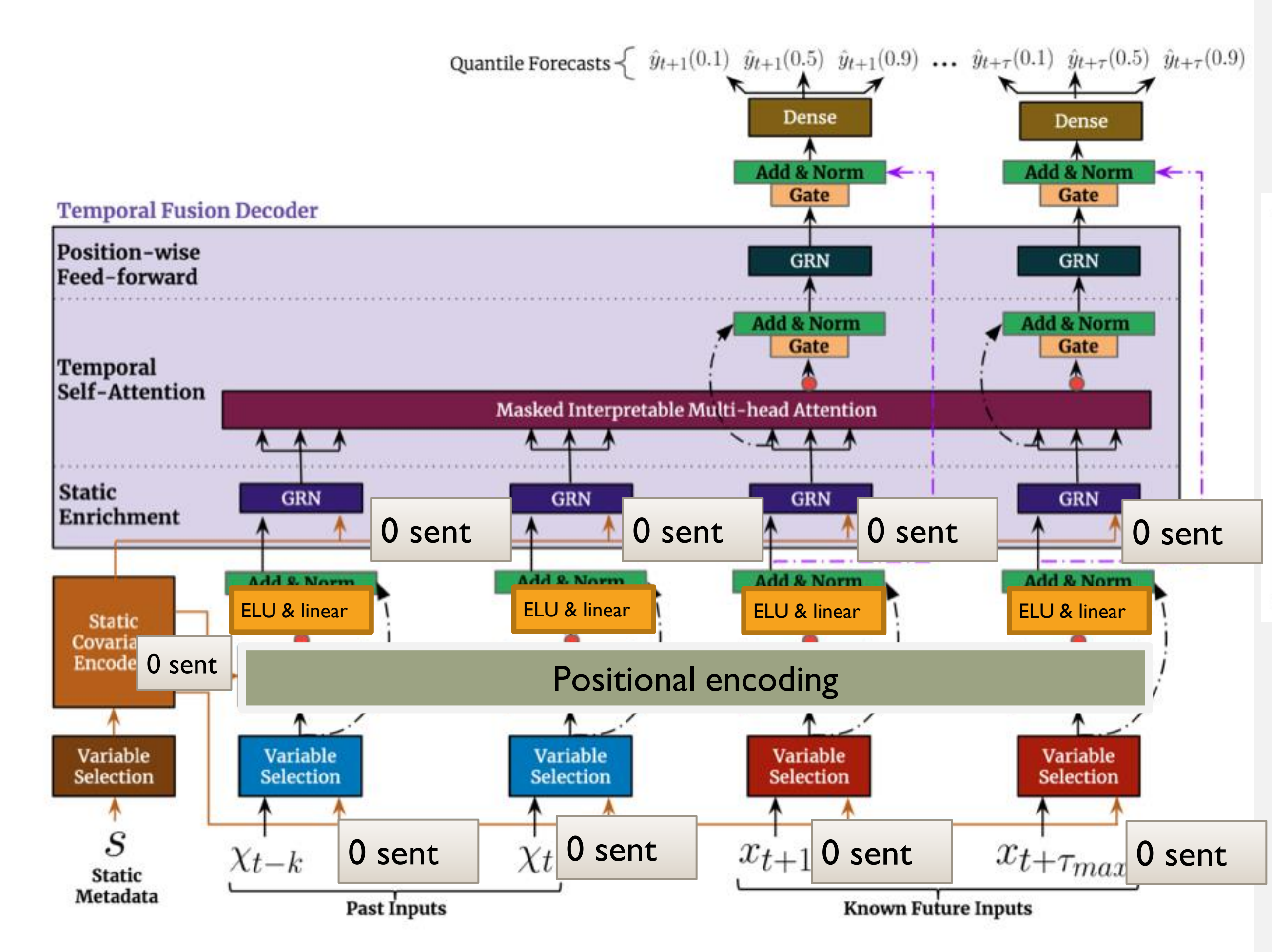
는 direct methods   cf. iterative methods

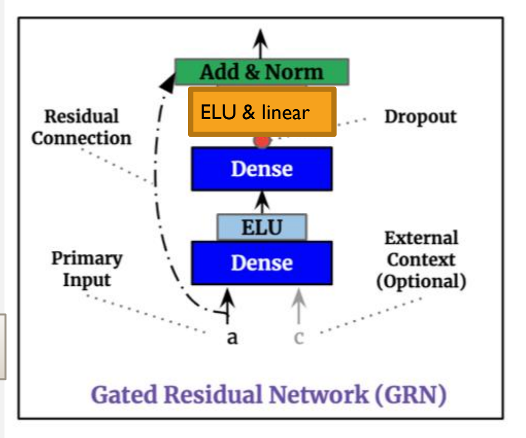
TFT 는 타 모델들에 비해 평균적으로 7% 낮은 P50 loss 와
9% 낮은 P90 loss 를 보여줌

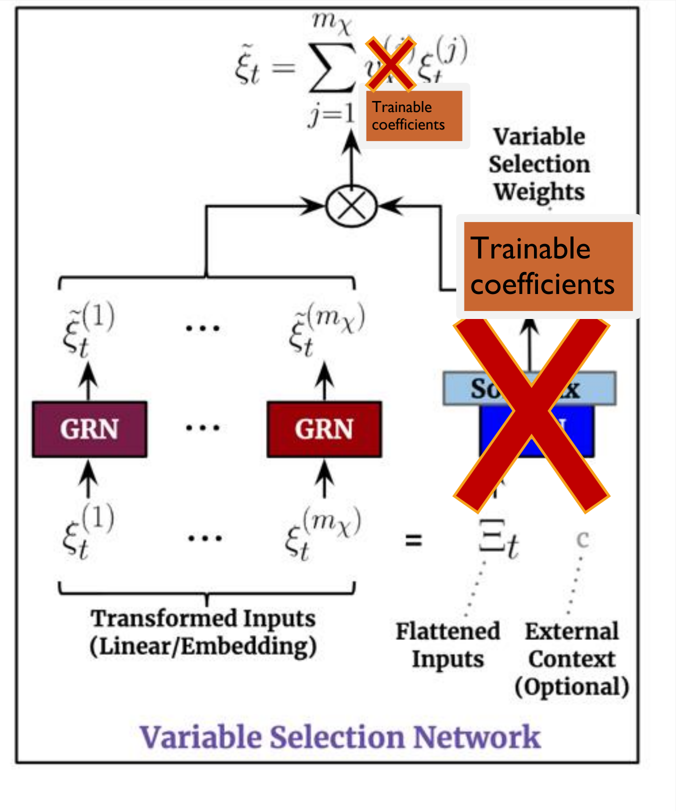
Iterative 방법을 사용한 핵심 모델 ConvTrans 의 경우 observed Input 등 다양하고 복잡한 데이터에서는 성능이 떨어짐

⇒ 즉 iterative methods 는 고정적인 input 값을 취해야 한다는 한계를 넘지 못하였음을 보여줌

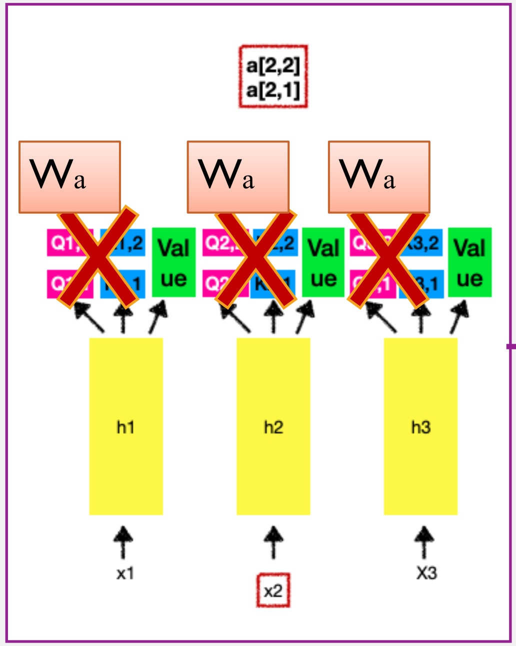
ABLATION ANALYSIS

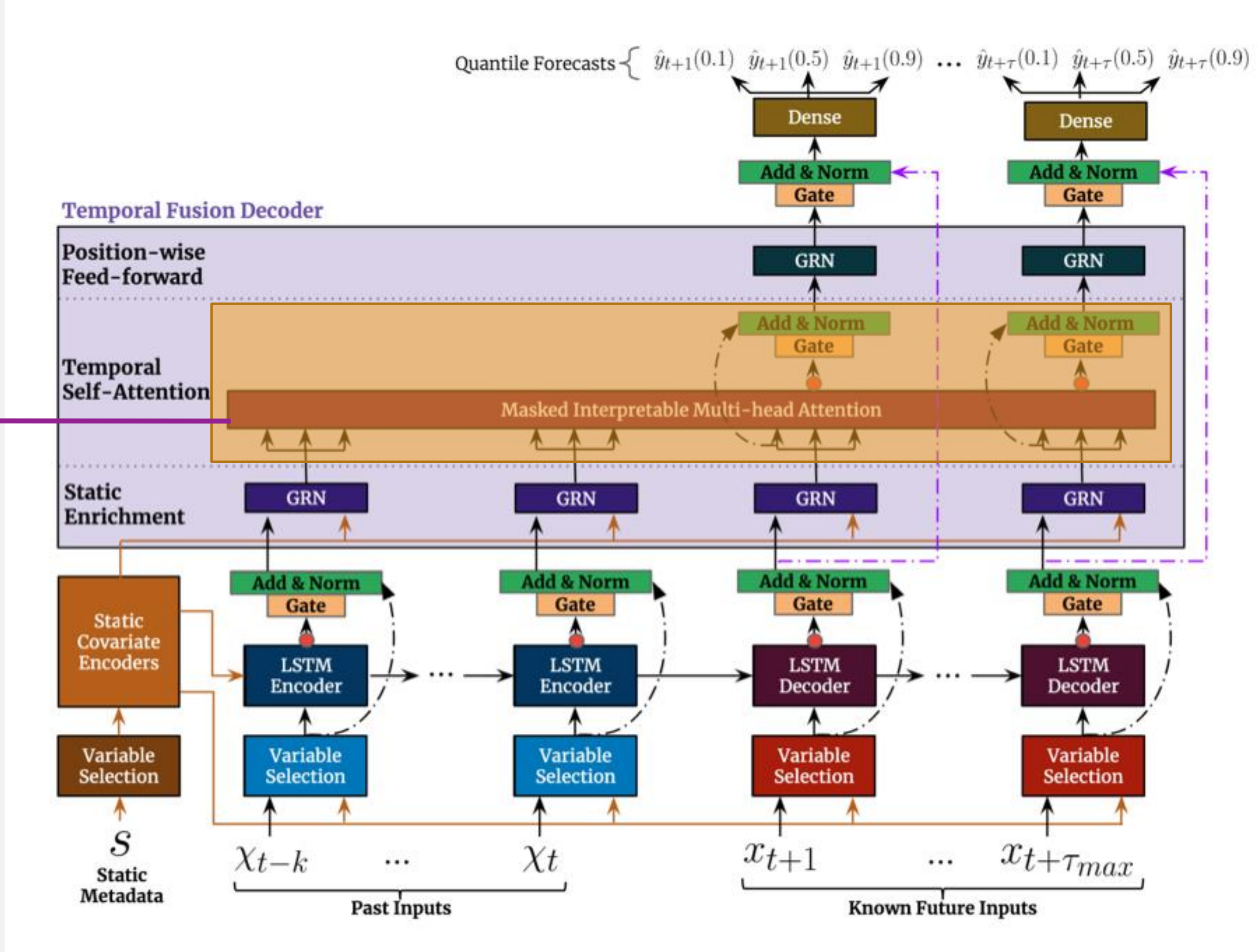






ABLATION ANALYSIS





ABLATION RESULTS

- Capturing temporal relationships, local processing ☆☆ : 비활성화 시켰더니 P90 loss 평균 6% 증가

- Local processing : 비활성화 시켰더니 `traffic,retail,volatility` 는 모두 악영향, `electricity` 는 오히려 P50 loss 높게 나옴 : Electricity data 의 경우 daily 단위로 seasonality 가 발견되기 때문에 direct attention to previous days > adjacent time steps
- Static covariate encoder , variable selection : 비활성화 시켰더니 P90 loss 평균 4.1% 증가, `electricity` 에 제일 영향을 많이 미친 것으로 파악
- Gating layer : 비활성화 시켰더니 P90 loss 평균 1.9% 증가, 노이즈가 많은 `volatility` 에 제일 영향을 많이 미친 것으로 파악

6. INTERPRETABILITY

6-1 VARIABLE IMPORTANCE

- 각 variables 의 확률 분포에서 (ex. t=1 일 때 weight, t=2 일 때 weight …) 10% , 50%, 90% 의 값을 추출하여 분석

remind

Variable Selection Weights

$$\tilde{\xi}_t = \sum_{j=1}^{m_\chi} v_{\chi_t}^{(j)} \xi_t^{(j)}$$

=> 각 timestep 별로 weights 총 j 개 생성

6-1 VARIABLE IMPORTANCE

Table 3: Variable importance for the Retail dataset. The 10^{th}, 50^{th} and 90^{th} percentiles of the variable selection weights are shown, with values larger than 0.1 highlighted in purple. For static covariates, the largest weights are attributed to variables which uniquely identify different entities (i.e. item number and store number). For past inputs, past values of the target (i.e. log sales) are critical as expected, as forecasts are extrapolations of past observations. For future inputs, promotion periods and national holidays have the greatest influence on sales forecasts, in line with periods of increased customer spending.

	10%	50%	90%
Item Num	**0.198**	**0.230**	**0.251**
Store Num	**0.152**	**0.161**	**0.170**
City	0.094	0.100	0.124
State	0.049	0.060	0.083
Type	0.005	0.006	0.008
Cluster	**0.108**	**0.122**	**0.133**
Family	0.063	0.075	0.079
Class	**0.148**	**0.156**	**0.163**
Perishable	0.084	0.085	0.088

(a) Static Covariates

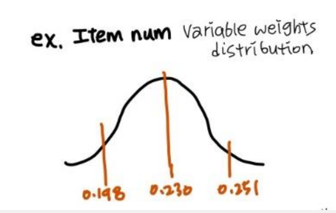
	10%	50%	90%
Transactions	0.029	0.033	0.037
Oil	0.062	0.081	0.105
On-promotion	0.072	0.075	0.078
Day of Week	0.007	0.007	0.008
Day of Month	0.083	0.089	0.096
Month	**0.109**	**0.122**	**0.136**
National Hol	**0.131**	**0.138**	**0.145**
Regional Hol	0.011	0.014	0.018
Local Hol	0.056	0.068	0.072
Open	0.027	0.044	0.067
Log Sales	**0.304**	**0.324**	**0.353**

(b) Past Inputs

	10%	50%	90%
On-promotion	**0.155**	**0.170**	**0.182**
Day of Week	0.029	0.065	0.089
Day of Month	**0.056**	**0.116**	**0.138**
Month	**0.111**	**0.155**	**0.240**
National Hol	**0.145**	**0.220**	**0.242**
Regional Hol	0.012	0.014	0.060
Local Hol	**0.116**	**0.151**	**0.239**
Open	0.088	0.095	0.097

(c) Future Inputs

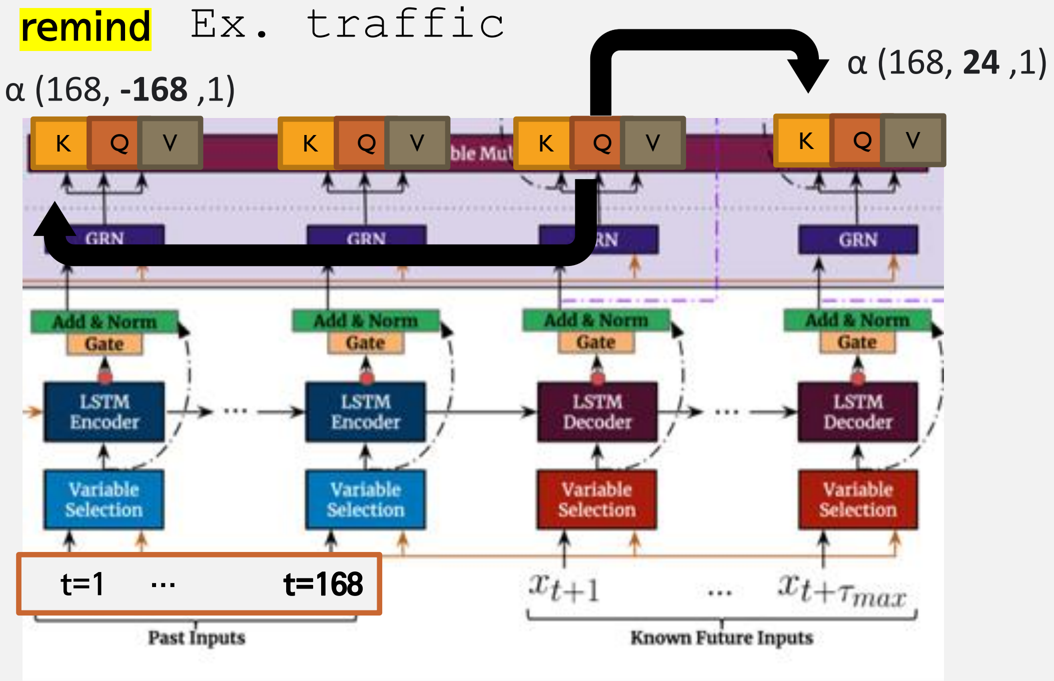
- TFT 는 예측에 실질적으로 유효한 변수들만을 추출하는 것으로 보임 (보라색)



6-2 VISUALIZING PERSISTENT TEMPORAL PATTERNS

(i.e. $\beta(t,\tau)$) can then be described as an attention-weighted sum of lower level features at each position n:

$$\beta(t,\tau) = \sum_{n=-k}^{\tau_{max}} \alpha(t,n,\tau)\,\tilde{\theta}(t,n), \qquad (27)$$



6-2 VISUALIZING PERSISTENT TEMPORAL PATTERNS

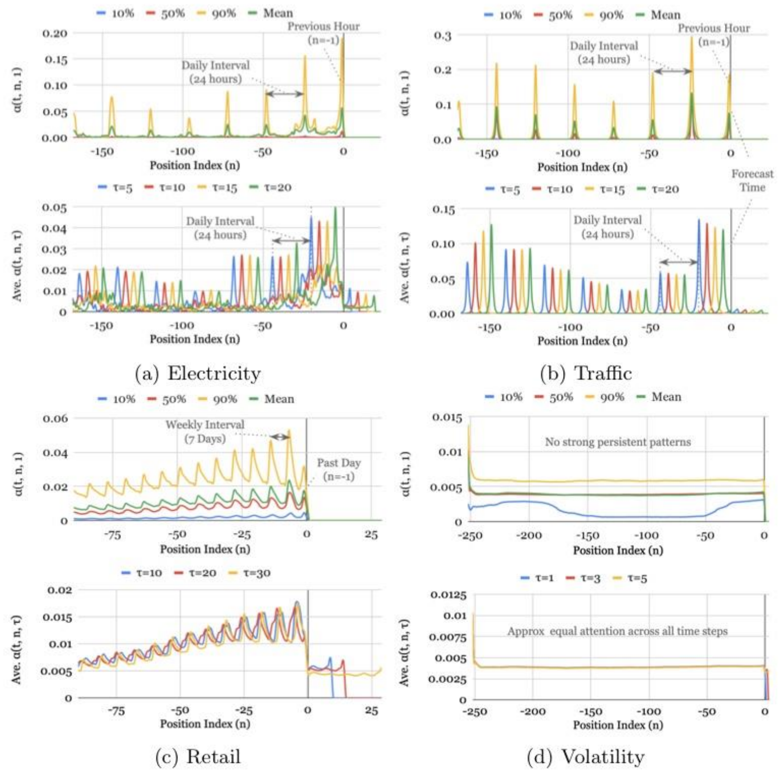


Figure 4: Persistent temporal patterns across datasets. Clear seasonality observed for the Electricity, Traffic and Retail datasets, but no strong persistent patterns seen in Volatility dataset. Upper plot – percentiles of attention weights for one-step-ahead forecast. Lower plot – average attention weights for forecast at various horizons.

- 세 데이터 모두 seasonal pattern 이 보임
- Attention spike (daily interval) – `Electricity, Traffic`
- Attention spike (weaker weekly interval) – `Retail`
- `Retail` 에서는 decaying trend pattern 이 나타남

6-3 IDENTIFYING REGIMES & SIGNIFICANT EVNETS

Firstly, for a given entity, we define the average attention pattern per forecast horizon as:

$$\bar{\alpha}(n,\tau) = \sum_{t=1}^{T} \alpha(t,j,\tau)/T, \tag{28}$$

and then construct $\bar{\boldsymbol{\alpha}}(\tau) = [\bar{\alpha}(-k,\tau),\ldots,\bar{\alpha}(\tau_{max},\tau)]^T$. To compare similarities

$\alpha(\square, \square, \square)$

Δ $t=?$

$\bar{\alpha}(1) = [\bar{\alpha}(-k,1) \ldots \bar{\alpha}(\tau_{max},1)]^T$

$t+1$ τ_{max}

t forecast horizon

$\bar{\alpha}(-k,1)$

$= \frac{1}{T}(\alpha(1,-k,1)+\alpha(2,-k,1)+\ldots$
$+\alpha(T,-k,1))$

→ 모든 timestep에서 한 시점 이후 Query와 -k window 이전 key의 attention score 평균

$\bar{\alpha}(\tau_{max},1)$

$= \frac{1}{T}(\alpha(1,\tau_{max},1)+\alpha(2,\tau_{max},1)+\ldots+$
$\alpha(T,\tau_{max},1))$

→ 모든 timestep에서 한 시점 이후 Query와 τ_{max} 시점 key의 attention score 평균

6-3 IDENTIFYING REGIMES & SIGNIFICANT EVNETS

Formula for measuring the overlap between discrete distributions

$$\kappa(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{1 - \rho(p, q)}, \tag{29}$$

where $\rho(\boldsymbol{p}, \boldsymbol{q}) = \sum_j \sqrt{p_j q_j}$ is the Bhattacharya coefficient [40] measuring the overlap between discrete distributions – with p_j, q_j being elements of probability vectors $\boldsymbol{p}, \boldsymbol{q}$ respectively. For each entity, significant shifts in temporal dynamics are then measured using the distance between attention vectors at each point with the average pattern, aggregated for all horizons as below:

모든 timestep 에서 특정 timestep t 에서

$$\text{dist}(t) = \sum_{\tau=1}^{\tau_{max}} \kappa\left(\bar{\boldsymbol{\alpha}}(\tau), \boldsymbol{\alpha}(t, \tau)\right) / \tau_{max}, \tag{30}$$

where $\boldsymbol{\alpha}(t, \tau) = [\alpha(t, -k, \tau), \ldots, \alpha(t, \tau_{max}, \tau)]^T$.

=> 공식에 넣고 평균 취함 !

6-3 IDENTIFYING REGIMES & SIGNIFICANT EVENTS

Volatility

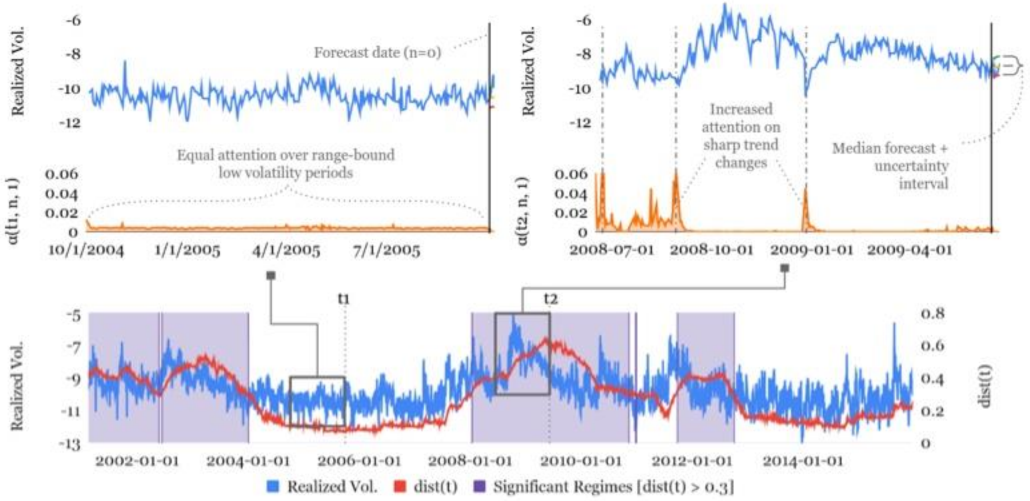


Figure 5: Regime identification for S&P 500 realized volatility. Significant deviations in attention patterns can be observed around periods of high volatility – corresponding to the peaks observed in dist(t). We use a threshold of dist(t) > 0.3 to denote significant regimes, as highlighted in purple. Focusing on periods around the 2008 financial crisis, the top right plot visualizes $\alpha(t, n, 1)$ midway through the significant regime, compared to the normal regime on the top left.

- High volatility 시기에는 trend 가 변함에 따라 attention 이 증가 (더 예민하게 반응)
- Low volatility 시기에는 attention 에 큰 변화가 없음

7. 결론 & 느낀 점

CONCLUSION

- 어텐션 메커니즘을 이용하여서 time series forecasting 을 하면 확실히 trend, seasonality 를 더 잘 포착할 수 있겠다 라는 생각이 듦
- Variable selections 의 경우 linear,softmax 조합보다 cnn 이 더 효과적일 수 있지 않을까 하는 생각이 듦
- 데이터의 종류에 따라서 적합한 모델이 다르게 나온 것을 보고 헬스케어 데이터에 적합한 모델은 어떠한 식으로 설계해야 할지 알아보고 싶어졌다 (ablation analysis)