TEMPORAL FUSION TRANSFORMERS FOR INTERPRETABLE MULTI-HORIZON TIME SERIES FORECASTING

2021.03.28

백지윤

목차

1. 연구 의의 및 목적

TFT VS RNN

ARCHITECTURE > GATING MECHANISMS

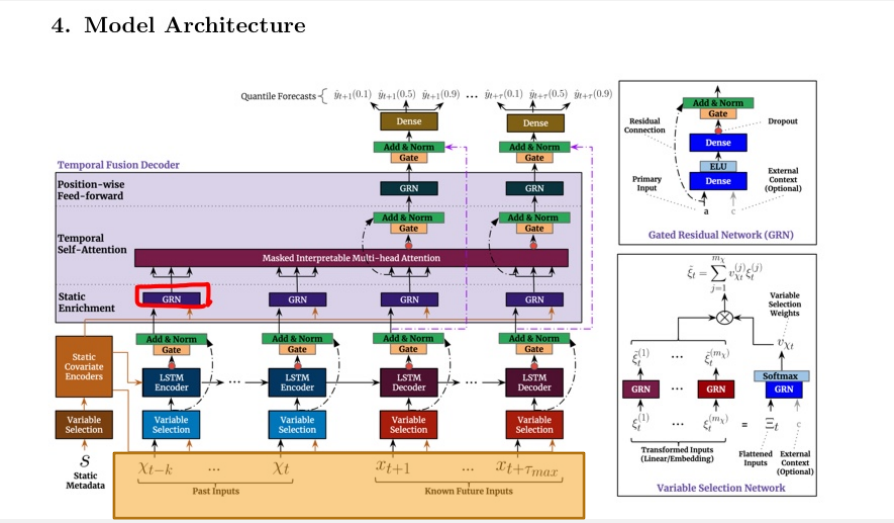
VARIABLE SELECTION NETWORKS

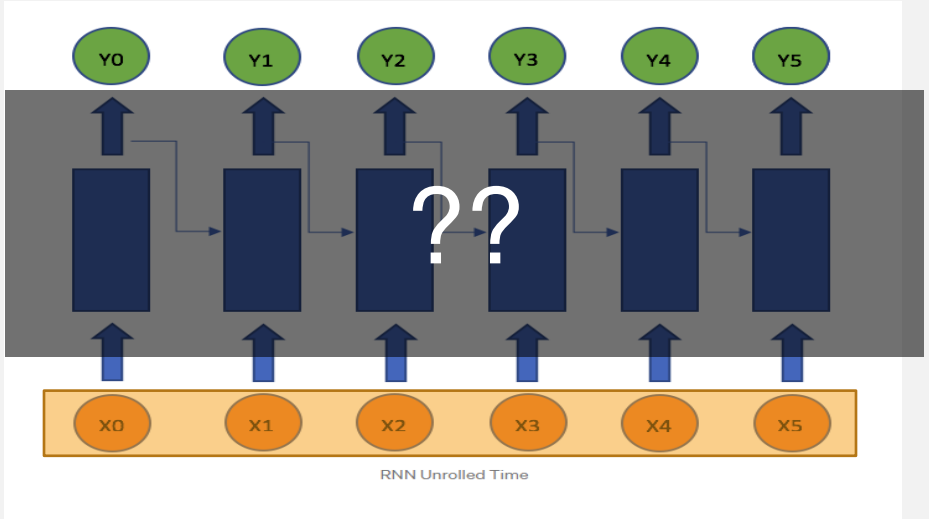
STATIC COVARIATE ENCODERS

ARCHITECTURE > GATING MECHANISMS ❌

VARIABLE SELECTION NETWORKS. ❌

STATIC COVARIATE ENCODERS ❌





input > Static covariates (contexts)
Observed inputs
Known inputs

input >
Observed inputs

연구 목적

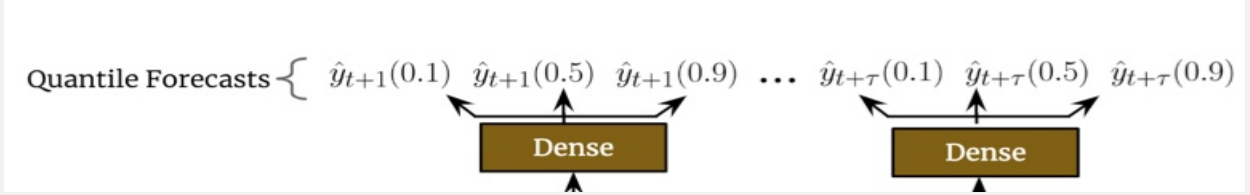
- Forecasting 에 영향을 줄 수 있는 보다 유연하고 풍부한 데이터를 모두 활용할 수 있는 모델을 만들겠다

- 모델 forecasting 도중 해당 시점의 연산에서 필수적인 레이어와 features 만을 필터링하여 사용하겠다 (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

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

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

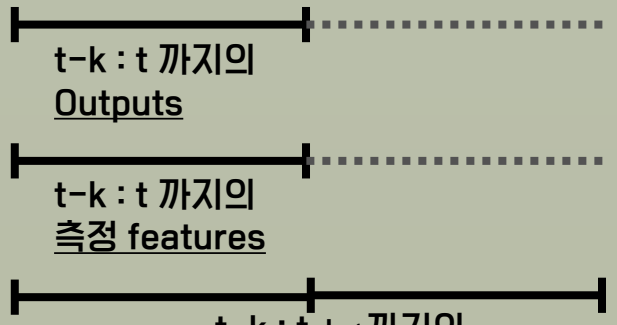
$$\hat{y}_i(q,t,\tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i)$$

현재 시점 - k시점 ; Window

현재 시점 (t)

t 시점 이후 예측값은?

이용할 features 4가지



Static covariates (메타 데이터)

=> features 를 이해할 수 있는 Context 로 같이 넣어줄 예정

3. 모델 구조

모델 핵심 요소 6가지

Gating Mechanisms

모델 구조

자유도 크게 / 자유도 작게

자유도 크게 / 자유도 작게

자유도 크게 / 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

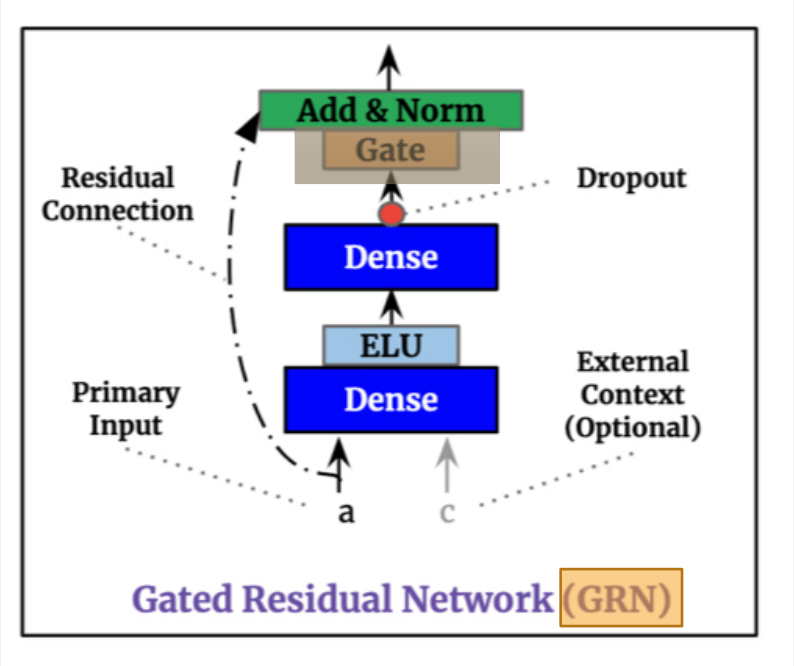
Temporal Fusion Decoder

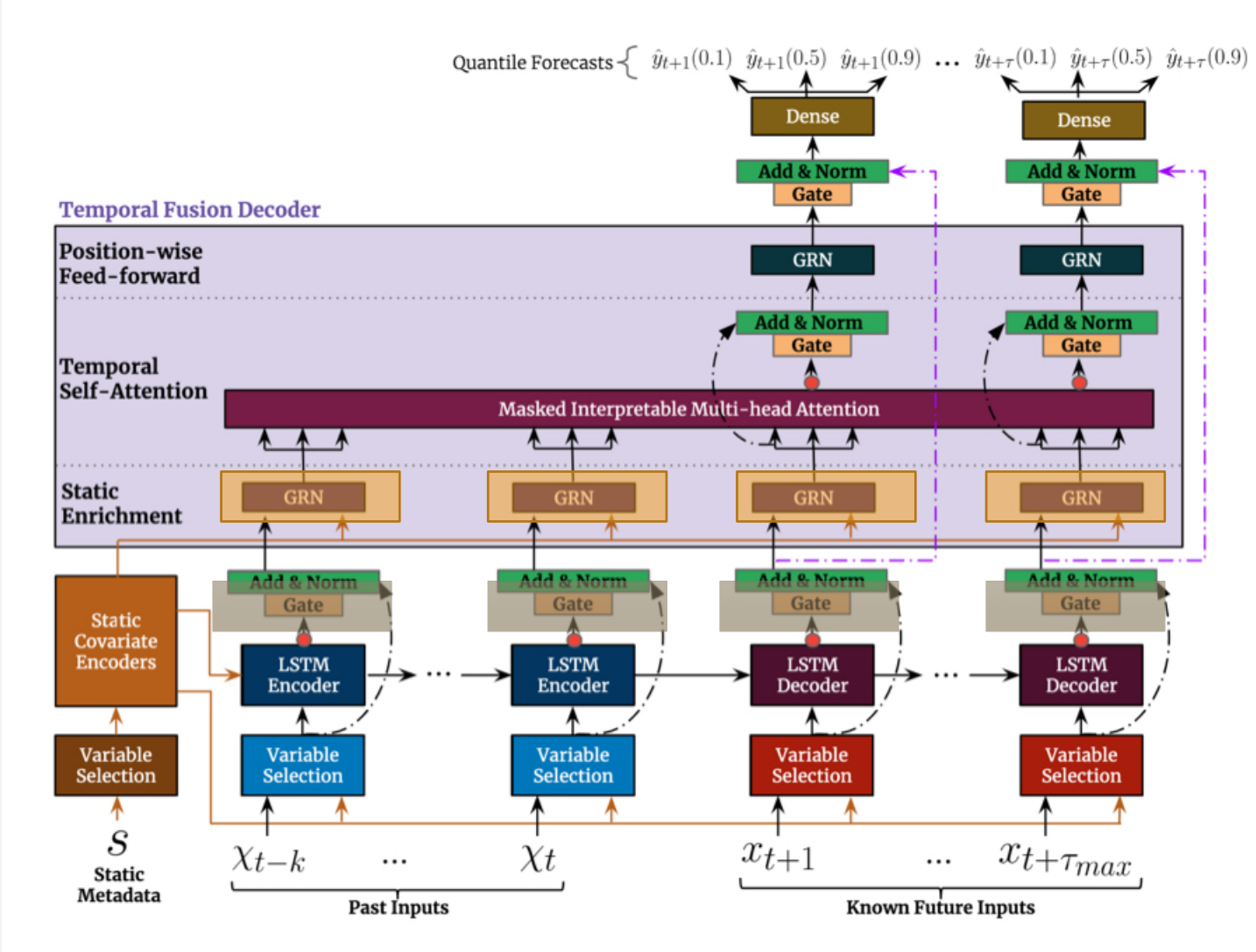
각 time step 의 장기간 상호관계 도출

Quantile Outputs

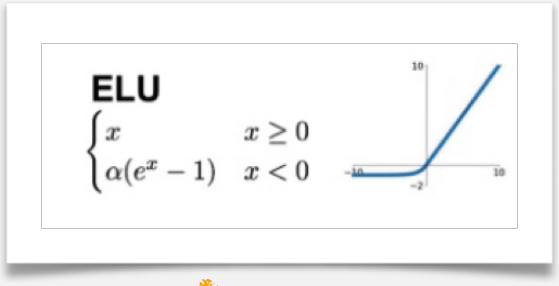
Gating Mechanisms = GRN layer

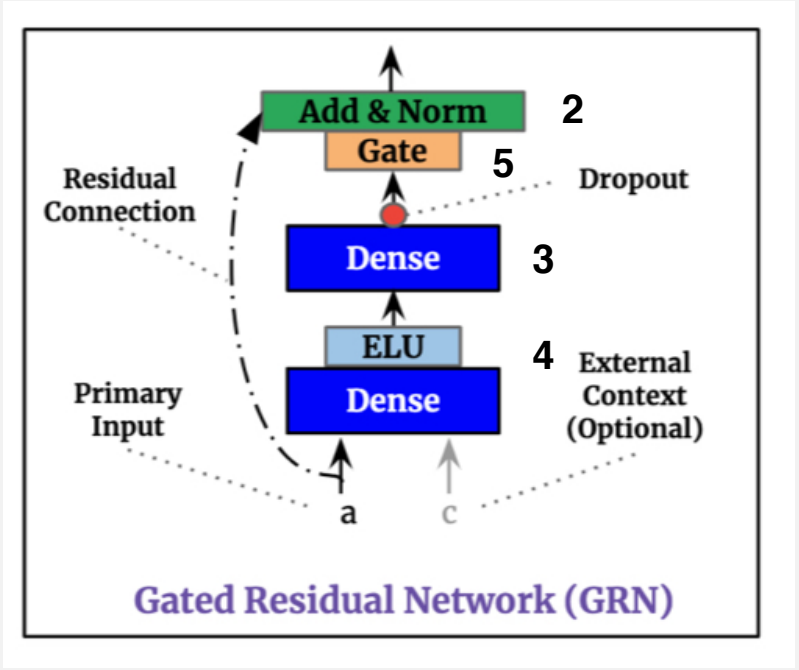
Gate 는 TFT 모델의 거의 모든 층에 사용 되는 핵심 테크닉
GRN layer 에도 gate 가 사용됨 !





Gating Mechanisms = GRN layer





$$\eta_2 = \mathrm{ELU}\left(W_{2,\omega}\, a + W_{3,\omega}\, c + b_{2,\omega}\right), \tag{4}$$

$$\eta_1 = W_{1,\omega}\, \eta_2 + b_{1,\omega}, \tag{3}$$

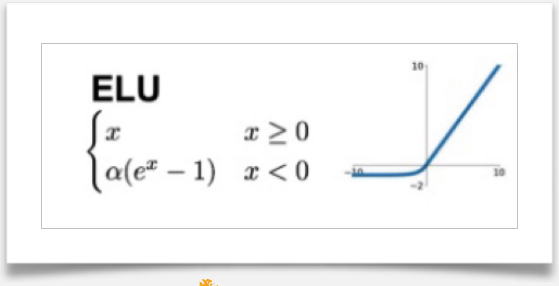
Gate

$$\mathrm{GLU}_\omega(\gamma) = \sigma(W_{4,\omega}\, \gamma + b_{4,\omega}) \odot (W_{5,\omega}\, \gamma + b_{5,\omega}), \tag{5}$$

Dropout (training)

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

Gating Mechanisms = GRN layer



(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

$$\eta_2 = \mathrm{ELU}\left(W_{2,\omega}\, a + W_{3,\omega}\, c + b_{2,\omega}\right), \quad (4)$$

$$\eta_1 = W_{1,\omega}\, \eta_2 + b_{1,\omega}, \quad (3)$$

$$\mathrm{GLU}_\omega(\gamma) = \sigma(W_{4,\omega}\, \gamma + b_{4,\omega}) \odot (W_{5,\omega}\, \gamma + b_{5,\omega}), \quad (5)$$

Gate

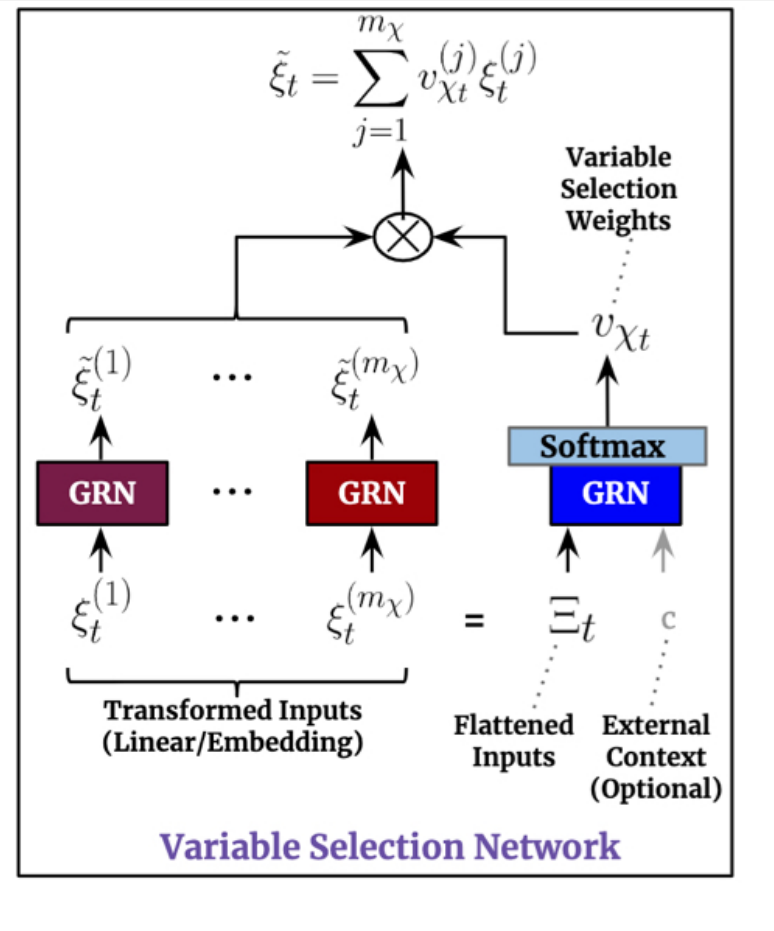
$$\mathrm{GRN}_\omega\,(a, c) = \mathrm{LayerNorm}\,(a + \mathrm{GLU}_\omega(\eta_1)), \quad (2)$$

(4) 가 모델 복잡도 결정

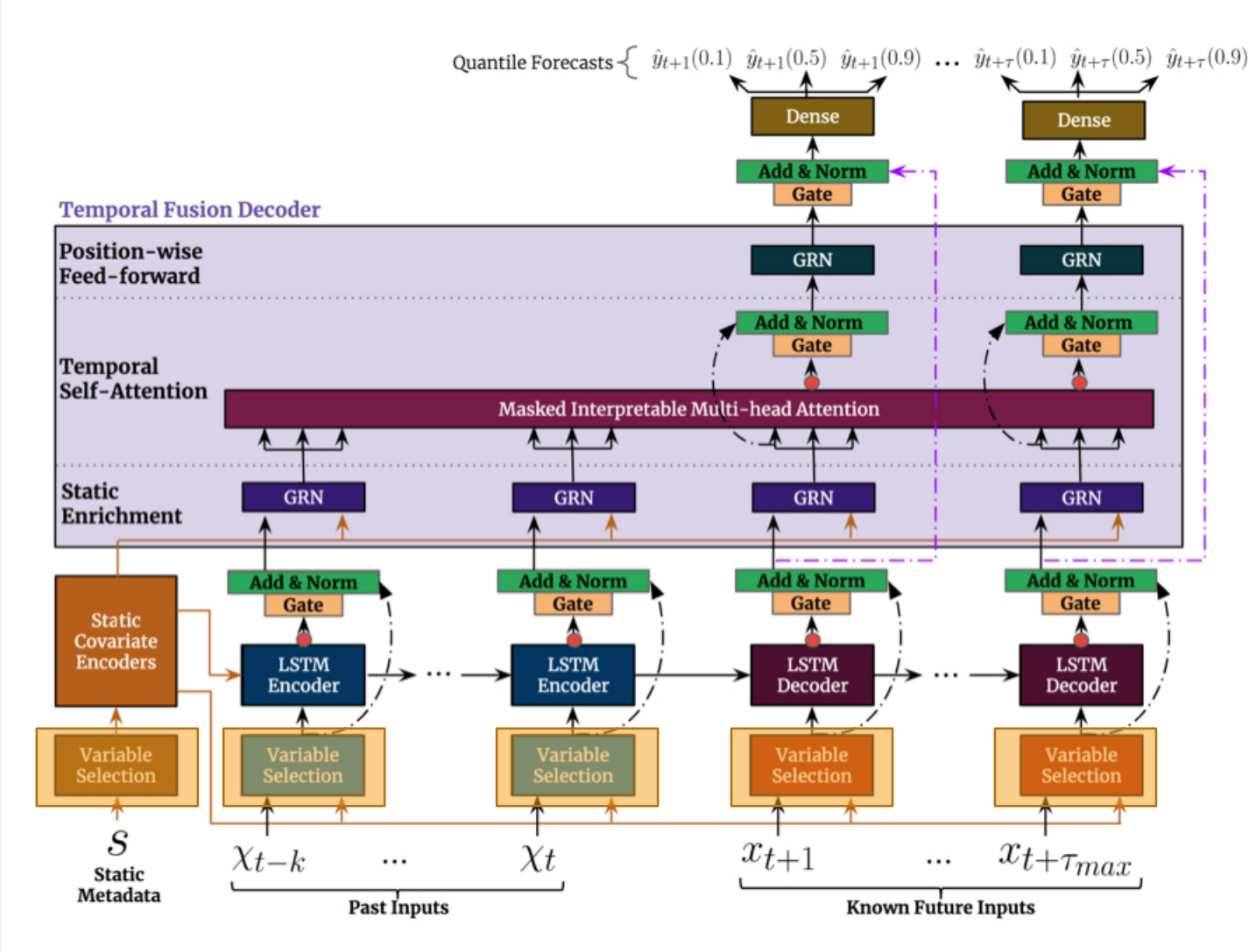
(5)-1 가 value scaling 역할

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

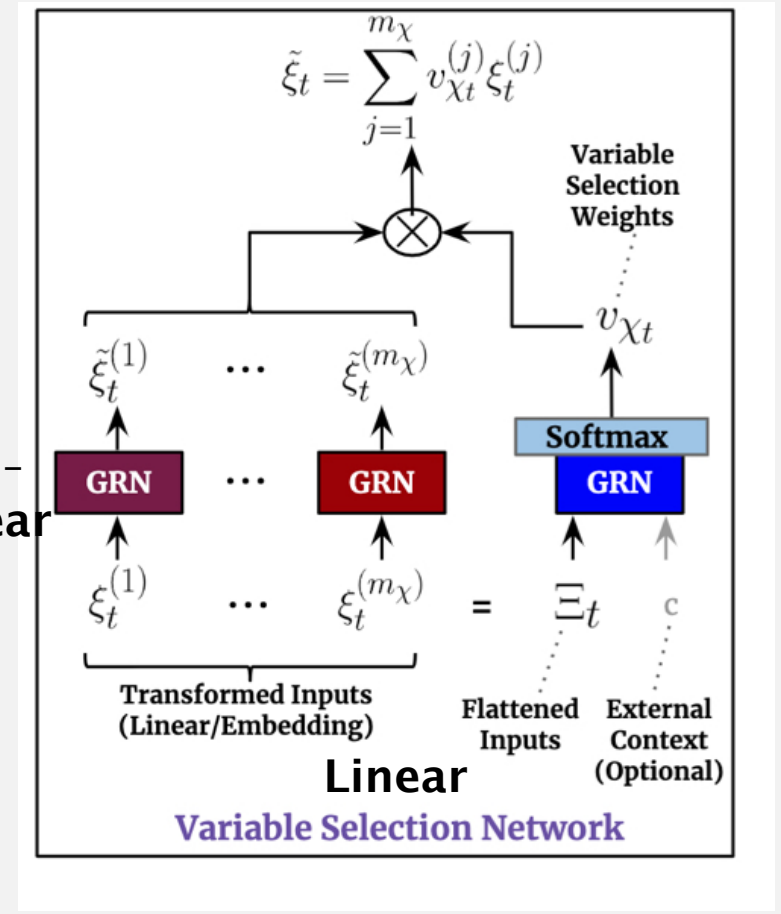
O → Entity embedding (D *Model* vector)

Continuous

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

100 →

Flattened Inputs



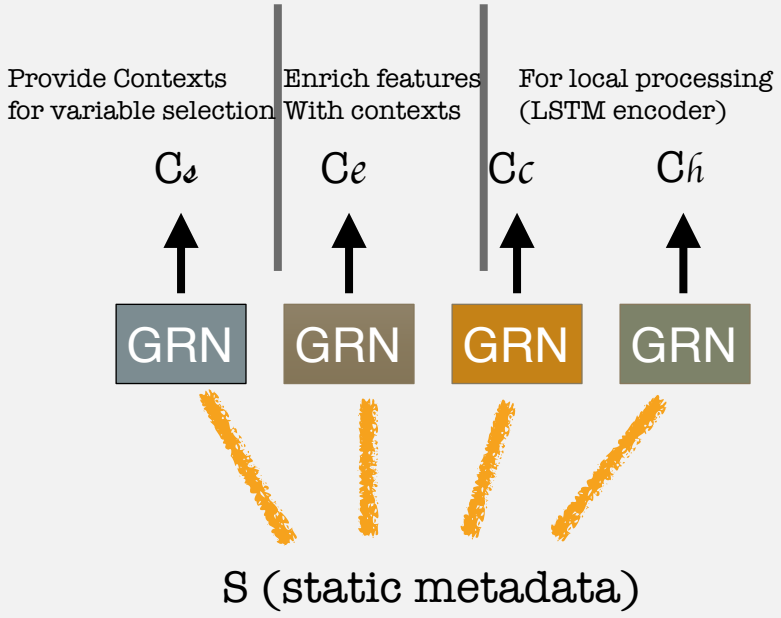
Variable Selection Weights

Feature 개수만큼 (j개)

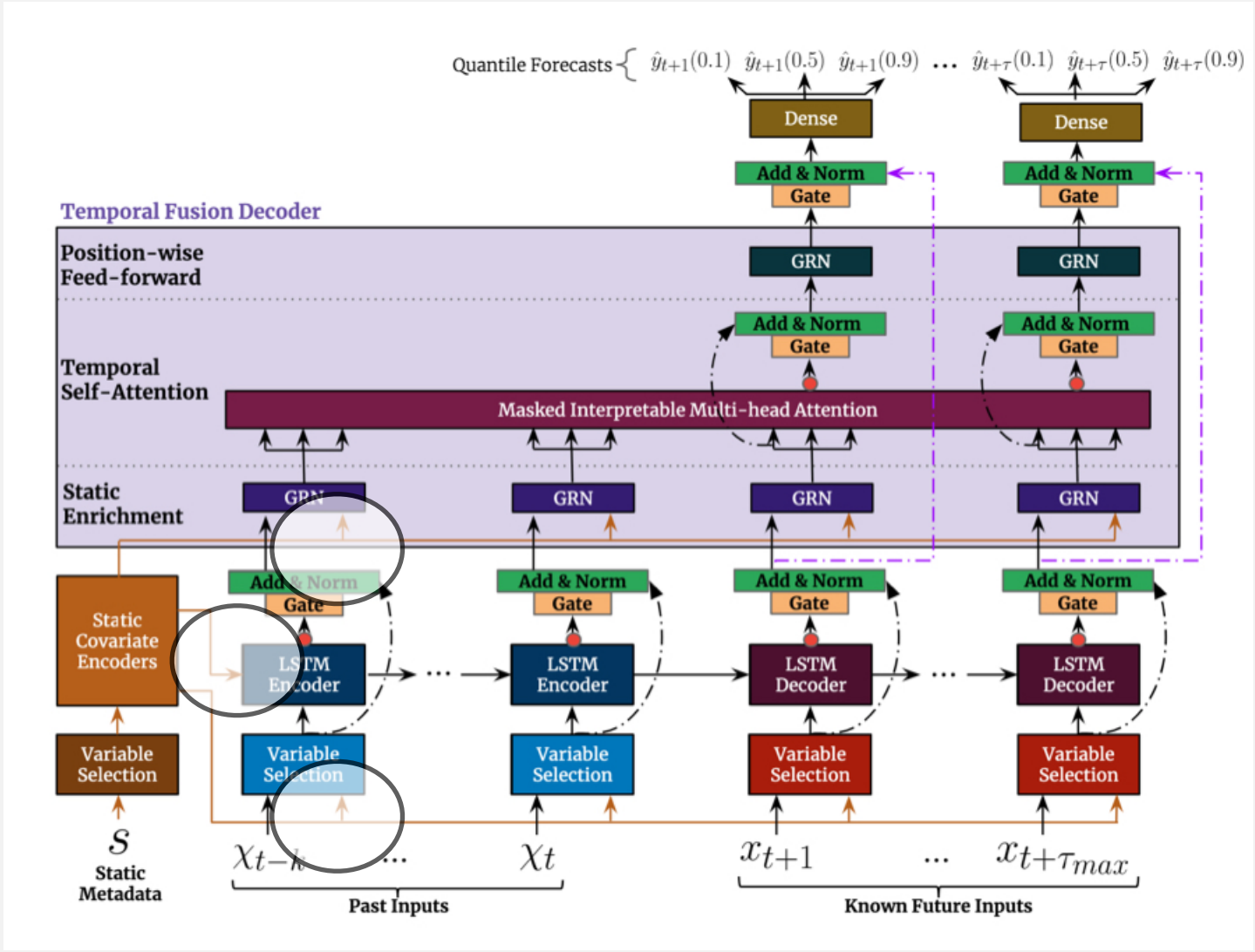
0.7 0.2 0.1

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

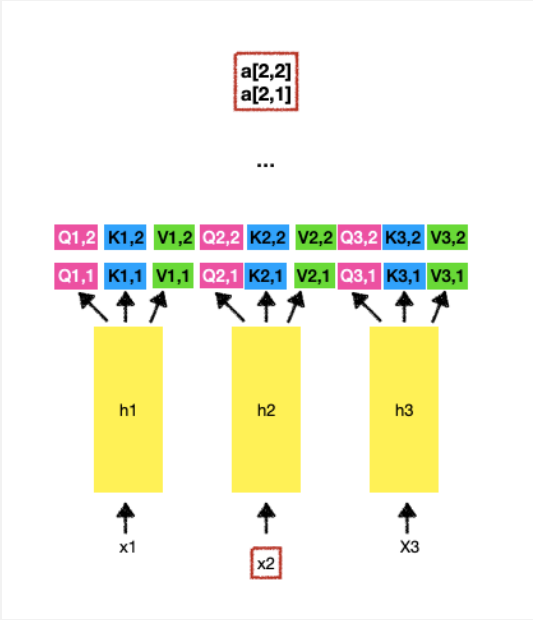


Interpretable Multi-Head Attention

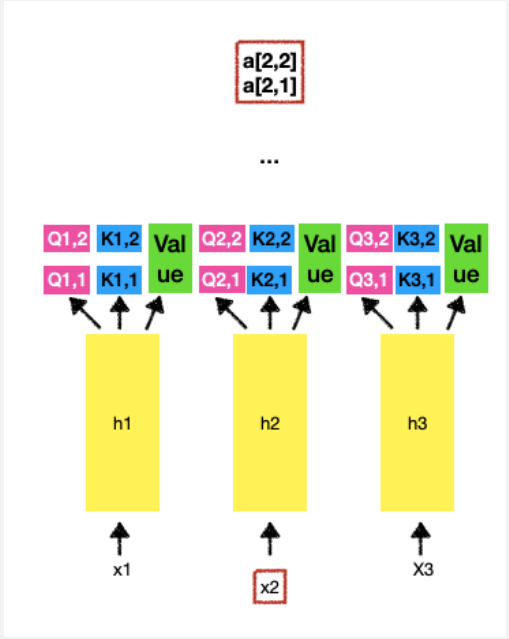
각 time step 의 장기간 상호관계 도출

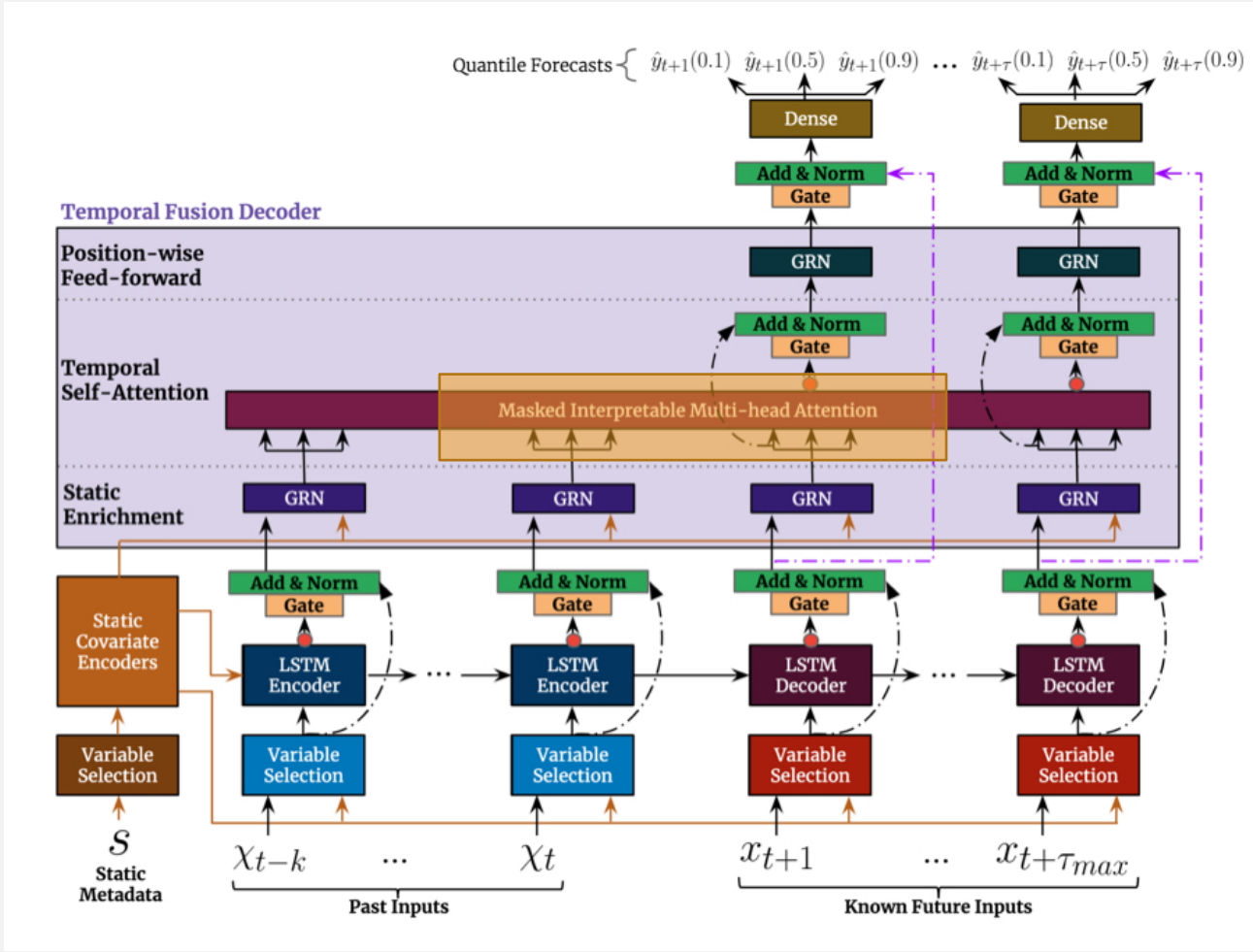
Multi-attention 아키텍처 그대로 갖고가되,
query,key,value 중 value 는 모든 head 에서 동일

원 Multi-head attention



TFT Multi-head attention





Interpretable Multi-Head Attention

각 time step 의 장기간 상호관계 도출

TFT Multi-head attention

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

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

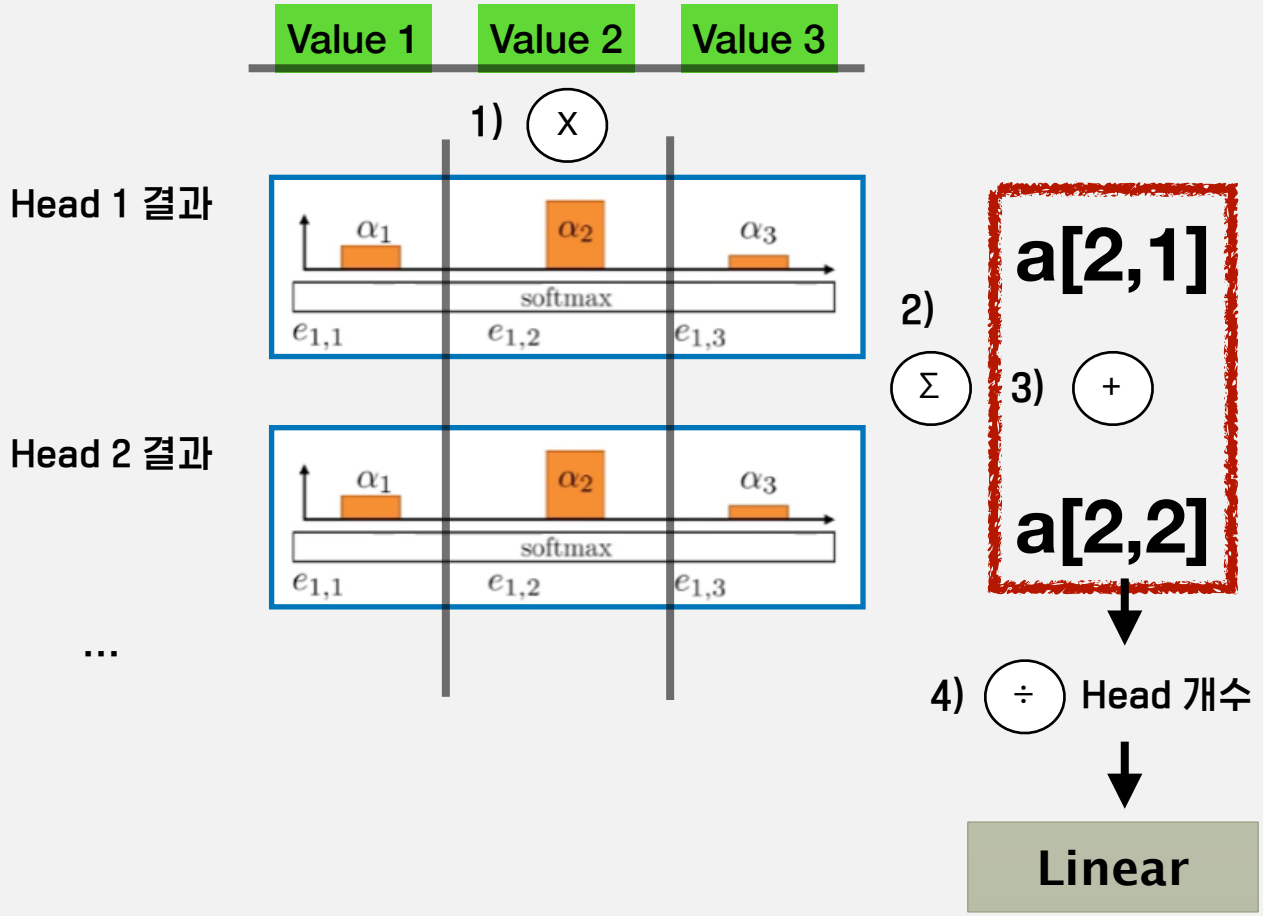
$$\text{InterpretableMultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \tilde{\boldsymbol{H}}\, \boldsymbol{W}_H, \tag{13}$$

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

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

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

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



ex. Timestep 2 의 어텐션 결과

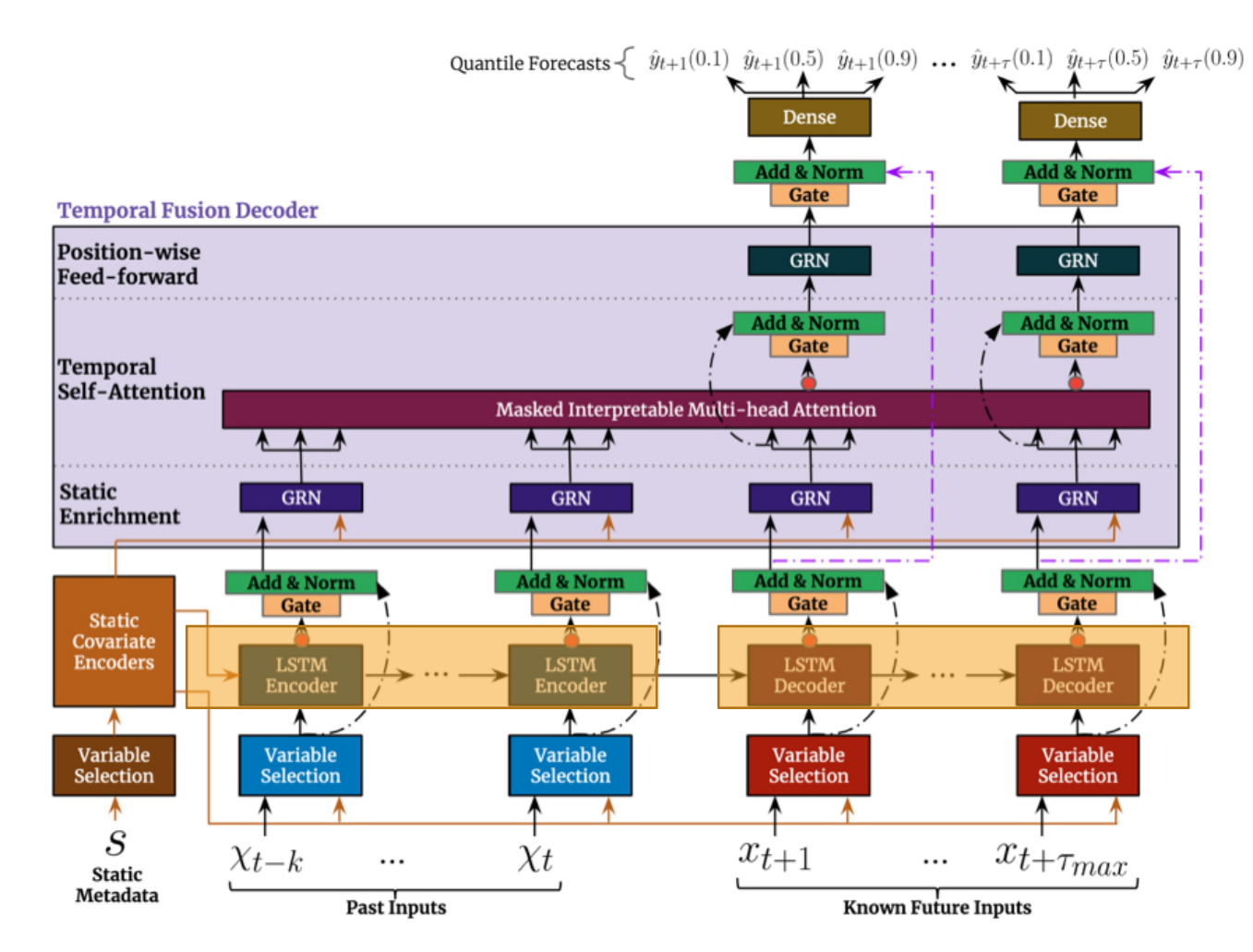
=> 앙상블 느낌 !

Temporal Fusion Decoder - (1)
Seq2Seq layer

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 에 들어가는 final inputs

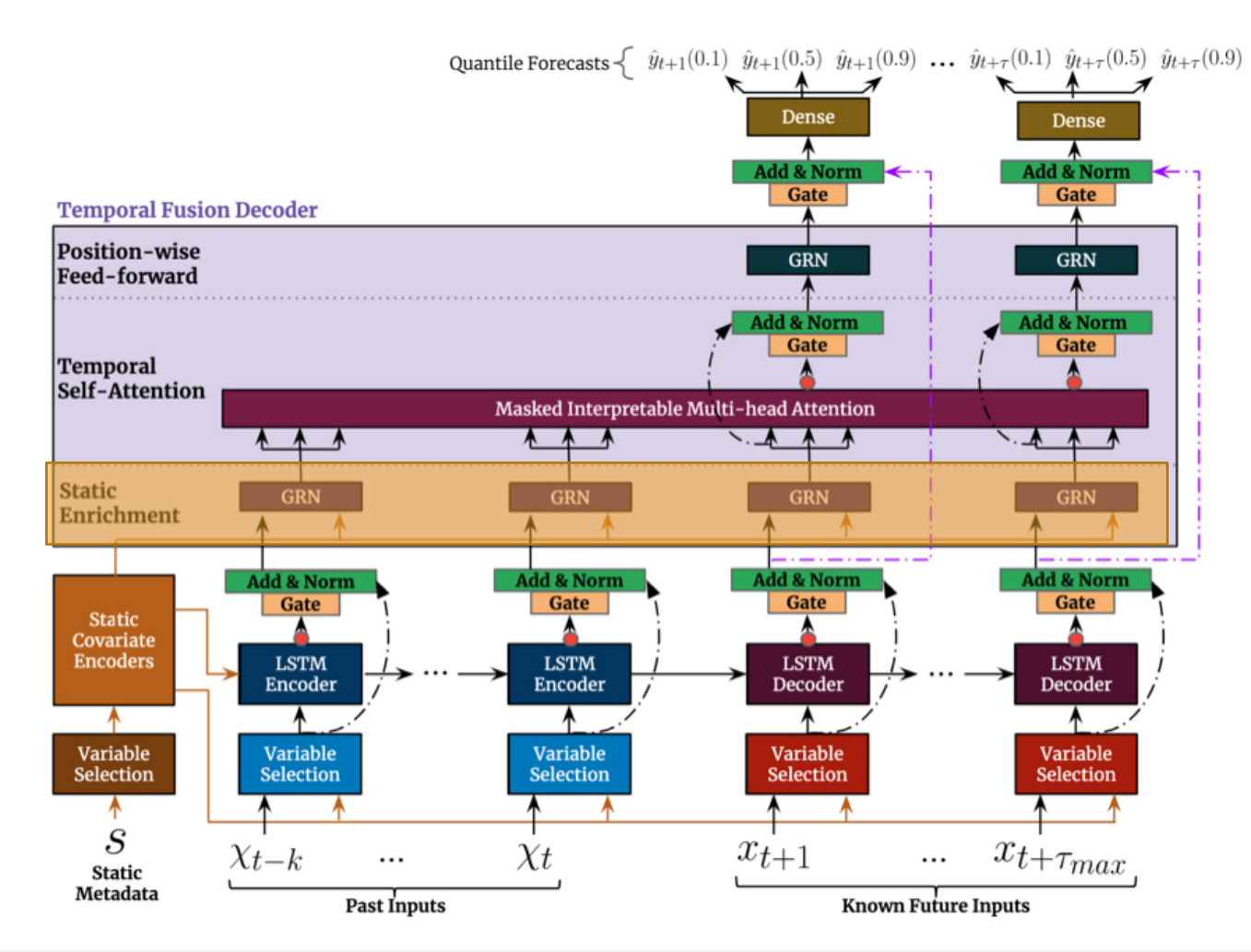


Temporal Fusion Decoder - (2)
Static Enrichment Layer

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 에 들어가는 final inputs

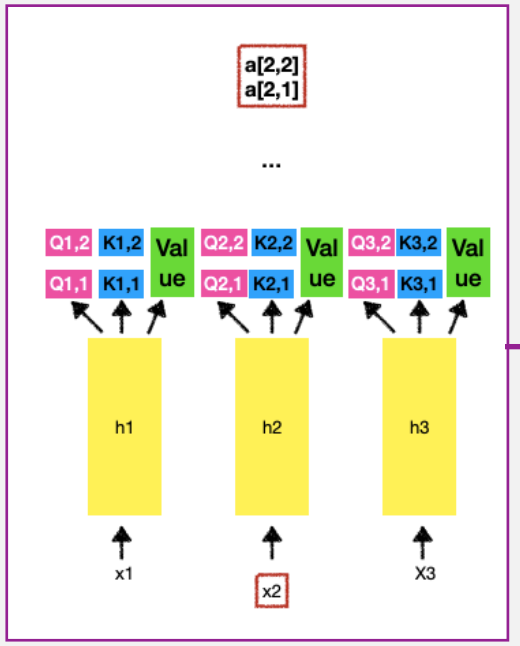


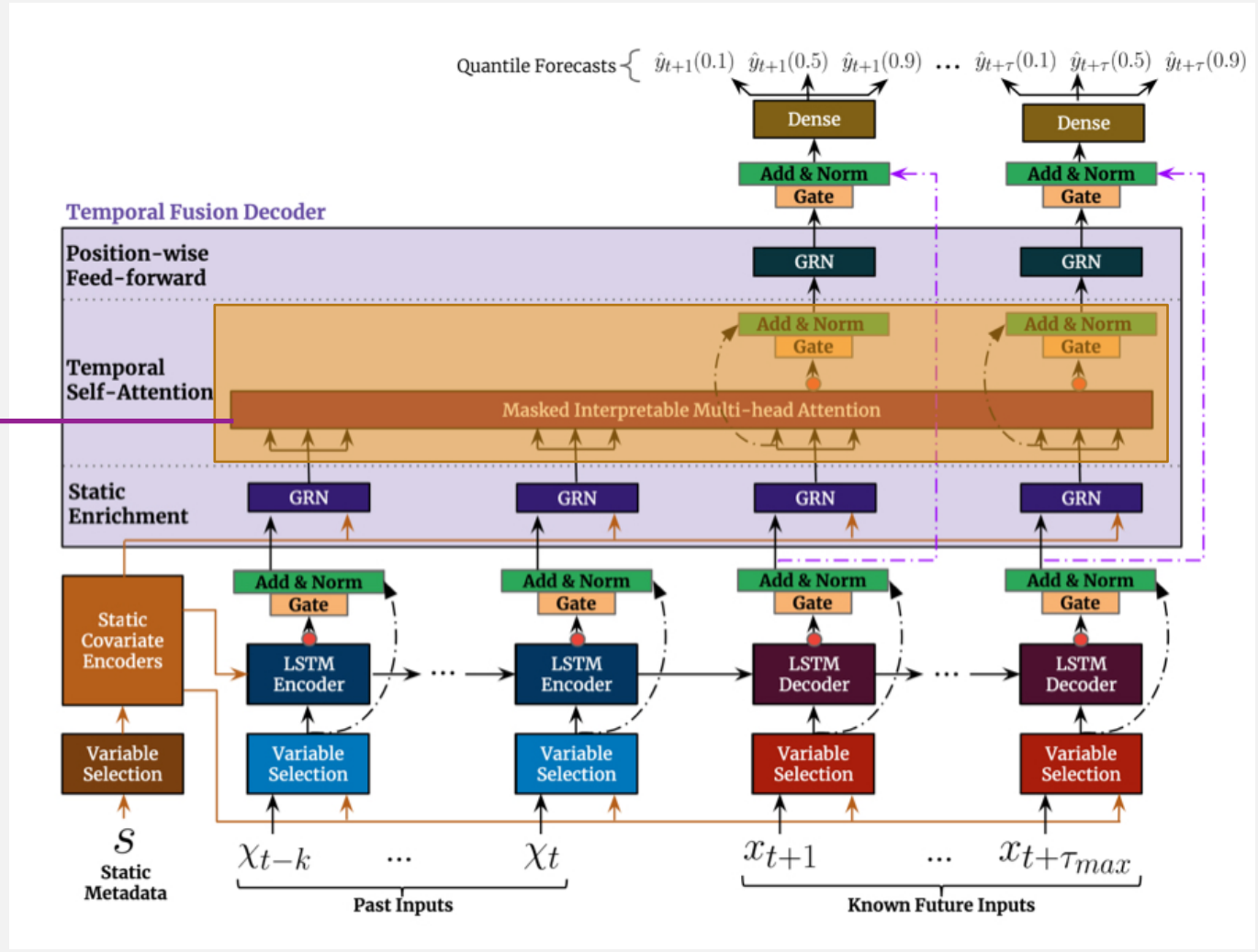
Temporal Fusion Decoder - (3)
Temporal Self-Attention Layer

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

↓

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

n > t

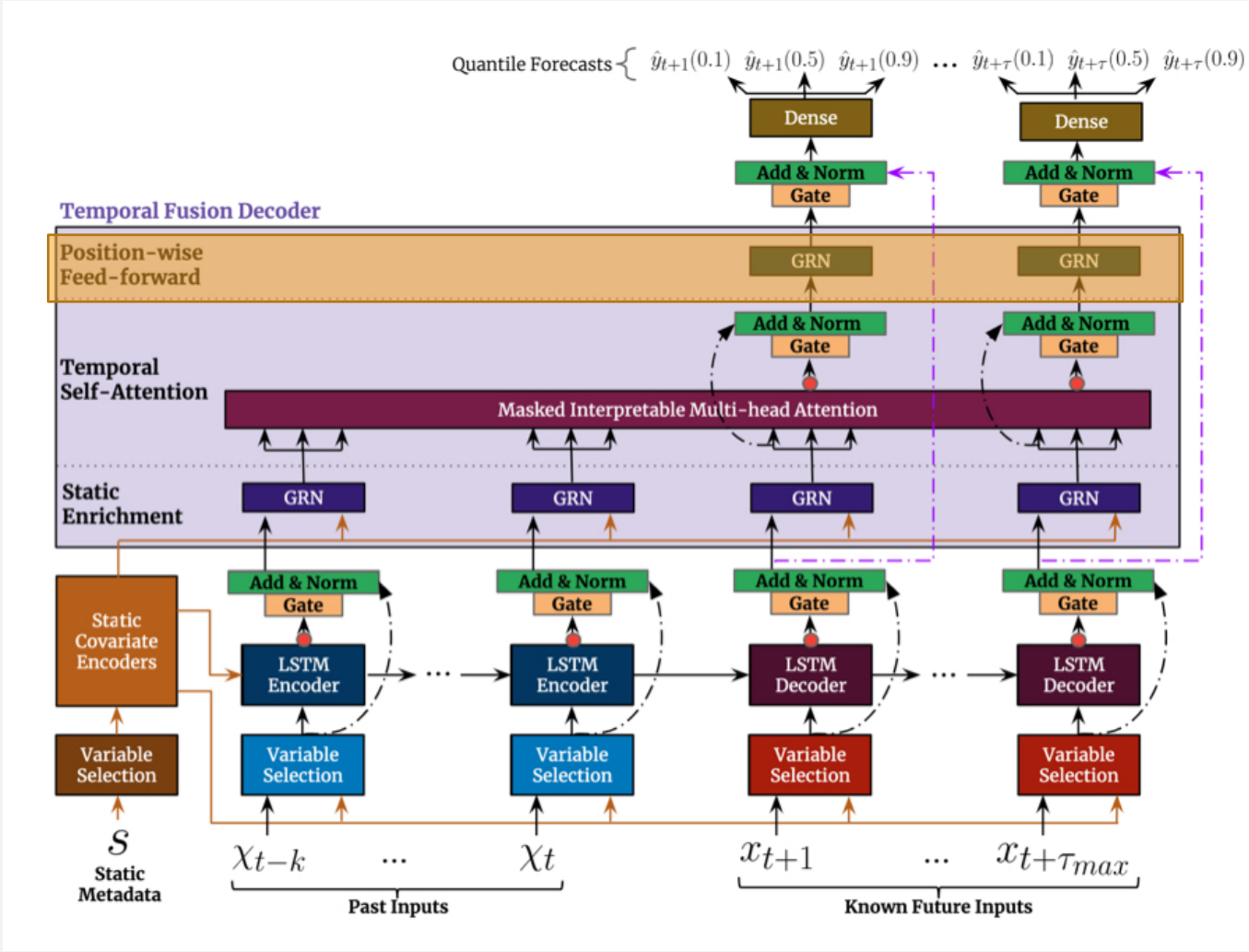
Temporal Fusion Decoder - (4)
Position-wise Feed-forward layer

n > t

non-linear 층 (GRN) 추가

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

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

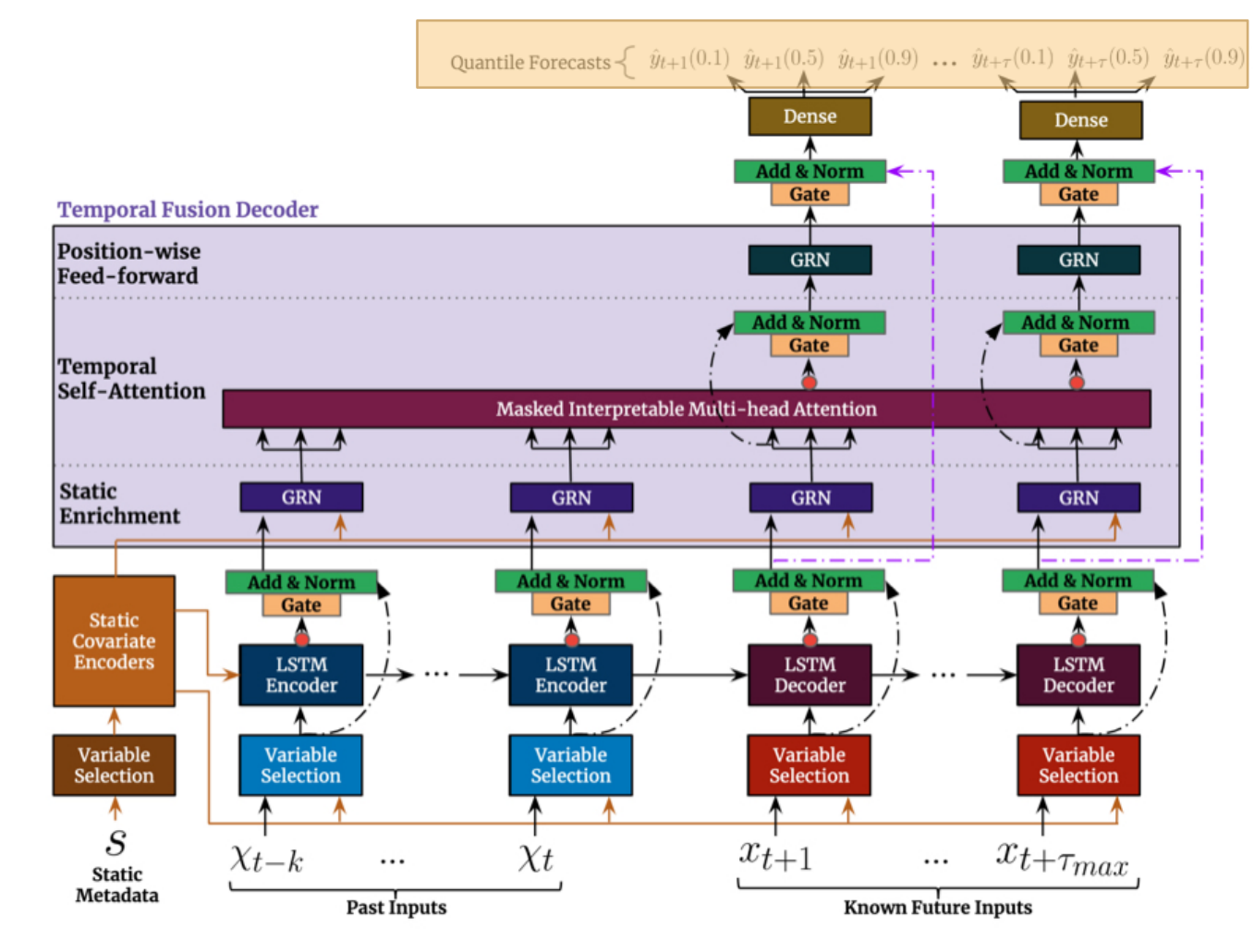


Quantile Outputs

최종 quantile 확률별 outputs 출력

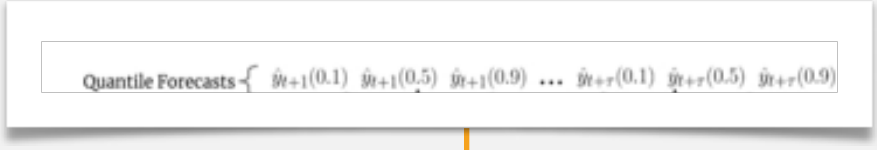
$$\hat{y}(q, t, \tau) = W_q \tilde{\psi}(t, \tau) + b_q, \qquad (23)$$

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



4. LOSS FUNCTION

LOSS FUNCTION



해당 데이터셋 內

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

{0.1,0.5,0.9}

예측할 개수

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

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**

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

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