**Question 1.** Answer

1.1 The expected complete data log likelihood (ECLL) :

$$\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})] = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x},\boldsymbol{z})]$$

we can rearrange

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}) &= \log p_\theta(\boldsymbol{x}) - \mathbb{E}_q \log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} \\
&= \log p_\theta(\boldsymbol{x}) - \mathbb{E}_q \log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{\frac{p_\theta(\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{x})}} \\
&= \log p_\theta(\boldsymbol{x}) - \mathbb{E}_q[\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{z},\boldsymbol{x}) + \log p_\theta(\boldsymbol{x})] \\
&= -\mathbb{E}_q[\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{z},\boldsymbol{x})] \\
&= \mathbb{E}_q[\log p_\theta(\boldsymbol{z},\boldsymbol{x})] + H(q)
\end{aligned}$$

Since the $H(q)$ is non-negative and $q(z|x)$ is fixed,hence, maximizing $ECLL = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x},\boldsymbol{z})] = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})]$ w.r.t $\theta$ is equal to maximize $\log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})$. It has the maximzum value where the value of $D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is 0.This means the maximizer of the ECLL coincides with that of the marginal likelihood only if $q(\boldsymbol{z}|\boldsymbol{x})$ perfectly matches $p(\boldsymbol{z}|\boldsymbol{x})$.

1.2

$$\begin{aligned}
\mathcal{L}(\theta,\phi;\boldsymbol{x}) &= \mathbb{E}_{q\phi}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})||p(\boldsymbol{z}))] \\
&= \mathbb{E}_{q_\phi}[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{z})} - \log q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) + \log q_(\boldsymbol{z}))] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{x},\boldsymbol{z}) - \log p(\boldsymbol{z}) - \log q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) + \log q_\phi(\boldsymbol{z}))] \\
&= \mathbb{E}_{q_\phi}[\log p_\theta(\boldsymbol{x},\boldsymbol{z}) - \log q_\phi(\boldsymbol{z} \mid \boldsymbol{x})] \\
&= \log p_\theta(\boldsymbol{x}) - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})
\end{aligned}$$

Hence,

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}) - \mathcal{L}(\theta,\phi;\boldsymbol{x}) &= D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}) \\
&= D_{\mathrm{KL}}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x})||p(\boldsymbol{z})) - D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) \\
&+ D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) - D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))
\end{aligned}$$

Since the $q_\phi$ is parametetized by a neural network, it means there is an additional constraint imposed by requiring that all the variational parameters lie in the range of the network. Within a specific parametric family, nothing is more general than freely optimizing the variational parameters $_i$. This cost is known as the amortization gap(last row in above equation). For a network with infinite capacity, this gap goes away, and we can be (only) as good as the free-form optimization setting. However that is not the case in any practical implementations, as all networks have finite capacity. Therefore, $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i)) \leq D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$

1.3 (a) In terms of bias in estimating the marginal likelihood via the ELBO, in the best case scenario (i.e. when both approaches are optimal within the respective families), The $D_{\mathrm{KL}}(q_i^*(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ has less bias comparing with $D_{\mathrm{KL}}(q_{\phi^*}(\boldsymbol{z}|\boldsymbol{x}_i)||p_\theta(\boldsymbol{z}|\boldsymbol{x}_i))$ because we model the $q_i^*$ for each observation and optimize them jointly. While with $q_{\phi^*}$, the family chosen is simpler than the $q_i^*$, it will invoke higher bias.

(b) In terms of the computational point of view (efficiency), we use $q_{\phi^*}$ to optimize the parameters of the neural network instead of the individual parameters of each observation. Hence the parameters won't depending on the number of observations. So, with the presence of massive data points, the $q_{\phi^*}$ approach is more efficient.

(c) In terms of memory (storage of parameters), same argument as (b), we specify our parameters of $q_{\phi^*}$ and do not rely on the instance numbers the instance numbers. So, we have could have fewer parameters when we have massive observations. Hence. having a maximizer $\phi^*$ of the ELBO requires less memory than $q_i$ for each $x_i$ since the number of parameters for the latter need to optimize grows (at least) linearly with the number of observations.

**Question 2** (5-5-5-5)**.** One way to enforce autoregressive conditioning is via masking the weight parameters.[1] Consider a two-hidden-layer convolutional neural network without kernel flipping, with kernel size $3 \times 3$ and padding size 1 on each border (so that an input feature map of size $5 \times 5$ is convolved into a $5 \times 5$ output). Define mask of type A and mask of type B as

$$(\boldsymbol{M}^A)_{::ij} := \begin{cases} 1 & \text{if } i < 2 \\ 1 & \text{if } i = 2 \text{ and } j < 2 \\ 0 & \text{elsewhere} \end{cases} \qquad (\boldsymbol{M}^B)_{::ij} := \begin{cases} 1 & \text{if } i < 2 \\ 1 & \text{if } i = 2 \text{ and } j \leq 2 \\ 0 & \text{elsewhere} \end{cases}$$

where the index starts from 1. Masking is achieved by multiplying the kernel with the binary mask (elementwise). Specify the receptive field of the output pixel that corresponds to the third row and the third column (index 33 of Figure 1 (Left)) in each of the following 4 cases :

| | | | | |
|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |
| 41 | 42 | 43 | 44 | 45 |
| 51 | 52 | 53 | 54 | 55 |

| | | | | |
|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |
| 41 | 42 | 43 | 44 | 45 |
| 51 | 52 | 53 | 54 | 55 |

FIGURE 1 – (Left) $5 \times 5$ convolutional feature map. (Right) Template answer.

1. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
2. If we use $\boldsymbol{M}^A$ for the first layer and $\boldsymbol{M}^B$ for the second layer.
3. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^A$ for the second layer.
4. If we use $\boldsymbol{M}^B$ for the first layer and $\boldsymbol{M}^B$ for the second layer.

Your answer should look like Figure 1 (Right).

**Answer 1.** 1.

| | | | | |
|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |
| 41 | 42 | 43 | 44 | 45 |
| 51 | 52 | 53 | 54 | 55 |

2.

| | | | | |
|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |
| 41 | 42 | 43 | 44 | 45 |
| 51 | 52 | 53 | 54 | 55 |

3.

| | | | | |
|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |
| 41 | 42 | 43 | 44 | 45 |
| 51 | 52 | 53 | 54 | 55 |

4

| | | | | |
|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 |
| 21 | 22 | 23 | 24 | 25 |
| 31 | 32 | 33 | 34 | 35 |
| 41 | 42 | 43 | 44 | 45 |
| 51 | 52 | 53 | 54 | 55 |

1. An example of this is the use of masking in the Transformer architecture.

**Question 3.** Answer

3.1 Based on the question, we have the following properties of the normalizing flows :

$$\boldsymbol{x} = F(\boldsymbol{u}), \text{where } \boldsymbol{u} \sim P_U(\boldsymbol{u}), \boldsymbol{u} = F^{-1}(\boldsymbol{x})$$

$$P_X(\boldsymbol{x}) = P_U(\boldsymbol{u})|\text{det}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}| = P_U(\boldsymbol{u})|\text{det}J_{F^{-1}}(\boldsymbol{x})|$$

$$P_U(\boldsymbol{u}) = P_X(\boldsymbol{x})|\text{det}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}| = P_X(\boldsymbol{x})|\text{det}J_F(\boldsymbol{u})|$$

$$P_{F^{-1}(X)}(\boldsymbol{u}) = P_X(F(\boldsymbol{u}))|\text{det}J_F(\boldsymbol{u})|$$

$$P_{F(U)}(\boldsymbol{x}) = P_U(F^{-1}(\boldsymbol{x}))|\text{det}J_{F^{-1}}(\boldsymbol{x})| = \frac{P_U(F^{-1}(\boldsymbol{x}))}{|\text{det}J_F(\boldsymbol{u})|}$$

We can first show that :

$$\begin{aligned} D_{KL}[P_X||P_{F(U)}] &= \int_{\boldsymbol{x}} P_X(\boldsymbol{x})\text{log}\frac{P_X(\boldsymbol{x})}{P_{F(U)}(\boldsymbol{x})}d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}} P_X(\boldsymbol{x})[\text{log}P_X(\boldsymbol{x}) - \text{log}P_{F(U)}(\boldsymbol{x})]d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}} P_X(\boldsymbol{x})[\text{log}P_U(\boldsymbol{u}) + \text{log}|\text{det}J_{F^{-1}}(\boldsymbol{x})| - \text{log}P_{F(U)}(\boldsymbol{x})]d\boldsymbol{x} \end{aligned}$$

Then we can show for the reverse KL divergence as :

$$\begin{aligned} D_{KL}[P_{F^{-1}(X)}||P_U] &= \int_u P_{F^{-1}(X)}(\boldsymbol{u})\log\frac{P_{F^{-1}(X)}(\boldsymbol{u})}{P_U(\boldsymbol{u})}d\boldsymbol{u} \\ &= \int_u P_{F^{-1}(X)}(\boldsymbol{u})[\log P_{F^{-1}(X)}(\boldsymbol{u}) - \log P_U(\boldsymbol{u})]d\boldsymbol{u} \\ &= \int_u P_{F^{-1}(X)}(\boldsymbol{u})[\log P_{F^{-1}(X)}(\boldsymbol{u}) - \log P_U(\boldsymbol{u})]dF^{-1}(\boldsymbol{x}) \\ &= \int_x P_{F^{-1}(X)}(\boldsymbol{u})[\log P_{F^{-1}(X)}(\boldsymbol{u}) - \log P_U(\boldsymbol{u})]|\det J_{F^{-1}}(\boldsymbol{x})|d\boldsymbol{x} \\ &= \int_x P_{F^{-1}(X)}(\boldsymbol{u})|\det J_{F^{-1}}(\boldsymbol{x})|[\log P_{F^{-1}(X)}(\boldsymbol{u}) - \log P_U(\boldsymbol{u})]d\boldsymbol{x} \\ &= \int_x P_X(F(\boldsymbol{u}))|\text{det}J_F(\boldsymbol{u})||\det J_{F^{-1}}(\boldsymbol{x})|[\log P_{F^{-1}(X)}(\boldsymbol{u}) - \log P_U(\boldsymbol{u})]d\boldsymbol{x} \\ &= \int_x P_X(F(\boldsymbol{u}))[\log P_{F^{-1}(X)}(\boldsymbol{u}) - \log P_U(\boldsymbol{u})]d\boldsymbol{x} \\ &= \int_x P_X(F(\boldsymbol{u}))[P_X(F(\boldsymbol{u}))|\text{det}J_F(\boldsymbol{u})| - \log P_U(\boldsymbol{u})]d\boldsymbol{x} \\ &= \int_x P_X(\boldsymbol{x})[\log P_X(\boldsymbol{x})| + \log|\det J_F(\boldsymbol{u})| - \log P_U(\boldsymbol{u})]d\boldsymbol{x} \\ &= \int_x P_X(\boldsymbol{x})[\log P_U(\boldsymbol{u}) + \log|\det J_{F^{-1}}(\boldsymbol{x})| + \log|\det J_F(\boldsymbol{u})| - \log P_U(\boldsymbol{u})]d\boldsymbol{x} \\ &= \int_x P_X(\boldsymbol{x})[\log P_U(\boldsymbol{u}) + \log|\det J_{F^{-1}}(\boldsymbol{x})| - \log\frac{P_U(\boldsymbol{u})}{|\det J_F(\boldsymbol{u})|}]d\boldsymbol{x} \end{aligned}$$

By replacing $\frac{P_U(\boldsymbol{u})}{|\det J_F(\boldsymbol{u})|} = P_{F(U)}(\boldsymbol{x})$, we have the above equation equal to

$$\int_{\boldsymbol{x}} P_X(\boldsymbol{x})[\log P_U(\boldsymbol{u})) + \log|\det J_{F^{-1}}(\boldsymbol{x})| - \log P_{F(U)}(\boldsymbol{x})]d\boldsymbol{x}$$

which is the same as the final equation from the $D_{KL}[P_X||P_{F(U)}]$

3.2 (a) For scenario 1), we would use the reverse KL divergence $D_{KL}[P_{F(U)}||P_X]$ as the objective to optimize. We can write the objective funtion as :

$$\mathcal{L} = D_{KL}[P_{F(U)}||P_X] = -E_{P_{F(U)}}[\log P_{F(U)} - \log P_X] \quad (1)$$
$$= -E_{p_U(\boldsymbol{u})}[\log p_U(\boldsymbol{u}) - \log|\det J_F(\boldsymbol{u})| - \log P_X(F(\boldsymbol{u}))]. \quad (2)$$

In order to minimize the reverse KL divergence as described above, we need to sample from the base distribution $P_U$ as well as compute and differentiate through the transformation $F$ and its Jacobian determinant and evaluation of $P_X$. It is not necessarily sample from $P_X$.

(b) For scenario 2), we would use the forward KL divergence $D_{KL}[P_X||P_{F(U)}]$ as the objective to optimize. The objective function for the forward KL divergence $D_{KL}[P_X||P_{F(U)}]$ can be written as follows :

$$\mathcal{L} = D_{KL}[P_X||P_{F(U)}] = -E_{P_X(\boldsymbol{x})}[\log P_{F(U)}] + const. \quad (3)$$
$$= -E_{P_X(\boldsymbol{x})}[\log p_U(\boldsymbol{u})(F^{-1}(\boldsymbol{x})) + \log|\det J_{F^{-1}}(\boldsymbol{x})|] + const. \quad (4)$$

Since we have samples from the data distribution $P_X(\boldsymbol{x})$ and we assume a set of samples $\{\boldsymbol{x}_n\}_{n=1}^N$ from $P_X(\boldsymbol{x})$, we can estimate $E_{P_X}$ by Monte Carlo with following :

$$\mathcal{L} \approx -\frac{1}{N}\sum_{n=1}^{N}[\log p_U(\boldsymbol{u})(F^{-1}(\boldsymbol{x})) + \log|\det J_{F^{-1}}(\boldsymbol{x})|] + const.$$

and then minimize the above Monte Carlo approximation of the KL divergence with stochastic gradient-based methods.Hence ,we don't need to evaluate $P_X(\boldsymbol{x})$.

**Question 4.** Answer

1. Maximizing $V(d, g)$ w.r.t $d$ implies : $d_{k+1} = d_k + \alpha \nabla_{d_k} V(d, g)$ ; Minimizing $V(d, g)$ w.r.t $g$ implies : $g_{k+1} = g_k - \alpha \nabla_{g_k} V(d, g)$.
   Since the $\nabla_{d_k} V(d, g) = g_k$ and $\nabla_{g_k} V(d, g) = d_k$,hence

$$[d_{k+1}, g_{k+1}]^\top = \begin{pmatrix} 1 & \alpha \\ -\alpha & 1 \end{pmatrix} \begin{pmatrix} d_k \\ g_k \end{pmatrix}$$

   Thus,

$$A = \begin{pmatrix} 1 & \alpha \\ -\alpha & 1 \end{pmatrix}$$

2. We know that a stationary point is a point on the surface of the graph (of the function) where all its partial derivatives are zero (equivalently, the gradient is zero).
   In this case, by differentiating, we get : $\begin{pmatrix} d_k \\ g_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Therefore the stationary points on this graph occur when $d_k = g_k = 0$,which is $d = g = 0$. Therefore the coordinates of the stationary point are (0,0).
3. Assume $\exists \lambda, s.t. |\lambda| \gneqq 1$, $\lambda + Sp(A)$, then $\exists x, s.t Ax = \lambda x$. we can see that

$$\| \begin{pmatrix} d_{k+1} \\ g_{k+1} \end{pmatrix} \| = \| A \begin{pmatrix} d_k \\ g_k \end{pmatrix} \| = |\lambda|^{k+1} \| \begin{pmatrix} d_0 \\ g_0 \end{pmatrix} \|$$

   Since $\lambda \gneqq 1$, when $k \to \infty$ the behaviour of $d_k$ and $g_k$ is diverging.
   Also, since the

$$Trace(A) = 1 + 1 = 2$$
$$det(A) = 1 + \alpha^2 > 1$$

   Hence, as we increasing the value of $\alpha$, it implies a faster divergence.

**Question 5.** Answer

5.1 Since,

$$\|W_p \boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2$$

=

$$(W_p \boldsymbol{f}_1 - \boldsymbol{f}_2)^\top (W_p \boldsymbol{f}_1 - \boldsymbol{f}_2)$$

=

$$\boldsymbol{f}_1^\top W_p^\top W_p \boldsymbol{f}_1 - \boldsymbol{f}_2^\top W_p \boldsymbol{f}_1 - \boldsymbol{f}_1^\top W_p^\top \boldsymbol{f}_2 + \boldsymbol{f}_2^\top \boldsymbol{f}_2$$

=

$$tr(W_p^\top W_p \boldsymbol{f}_1^\top \boldsymbol{f}_1) - tr(W_p \boldsymbol{f}_1 \boldsymbol{f}_2^\top) - tr(W_p^\top \boldsymbol{f}_2 \boldsymbol{f}_1^\top) + tr(\boldsymbol{f}_2^\top \boldsymbol{f}_2)$$

Since $F_1 = F_2 = \mathbb{E}\left[\boldsymbol{f}_1 \boldsymbol{f}_1^{\mathrm{T}}\right] = W(X + X')W^{\mathrm{T}}$ and $F_{12} = F_{21}^T = \mathbb{E}\left[\boldsymbol{f}_1 \boldsymbol{f}_2^{\mathrm{T}}\right] = WXW^{\mathrm{T}}$.
Hence, we have

$$tr(W_p^\top W_p \boldsymbol{f}_1^\top \boldsymbol{f}_1) - tr(W_p \boldsymbol{f}_1 \boldsymbol{f}_2^\top) - tr(W_p^\top \boldsymbol{f}_2 \boldsymbol{f}_1^\top) + tr(\boldsymbol{f}_2^\top \boldsymbol{f}_2)$$

=

$$\frac{1}{2} tr(W_p^\top W_p F_1) - tr(W_p F_{12}) - tr(W_p^\top F_{12}) + tr(F_2)$$

Thus,

$$J(W, W_p) = \frac{1}{2} \mathbb{E}_{x_1, x_2}\left[\|W_p \boldsymbol{f}_1 - \text{Stop-Grad}(\boldsymbol{f}_2)\|_2^2\right]$$

=

$$J(W, W_p) = \frac{1}{2}\left[\text{tr}(W_p^{\mathrm{T}} W_p F_1) - \text{tr}(W_p F_{12}) - \text{tr}(F_{12} W_p) + \text{tr}(F_2))\right]$$

5.2

$$\dot{W}_p = -\frac{\partial J}{\partial W_p} = -\frac{1}{2}(2W_p F_1 - F_{12}^\top - F_{12}^\top) = -W_p F_1 + F_{12}^\top$$

5.3 Since we don't have stop gradient and $W$ is the weight for the target network, then we have additional terms. With $\tilde{W}_p(t) = W_p(t) - I_{n_2}$ and we have :

$$\dot{W} = -\frac{\partial J}{\partial W}$$

=

$$-W_p^\top W_p W(X + X') + (W_p^\top + W_p)WX - W(X + X')$$

=

$$-(W_p^\top W_p + I)WX' - (W_p^\top W_p - W_p^\top - W_p + I)WX$$

=

$$-(W_p^\top W_p + I)WX' - (W_p - I)^\top (W_p - I)WX$$

=

$$-(W_p^\top W_p + I)WX' - \tilde{W}_p{}^\top \tilde{W}_p WX$$

with $vec(AXB) = (B^T \otimes A)vec(X)$ and we have :

$$\frac{d}{dt}vec(W) = -[X' \otimes \left(W_p^T W_p + I\right) + X \otimes \tilde{W}_p^T \tilde{W}_p]vec(W)$$

Hence

$$\frac{d}{dt}\text{vec}(W(t)) = -H(t)\text{vec}(W(t))$$

if $\inf_{t\geq 0} \lambda_{min}(H(t)) \geq \lambda_0 > 0$, then apply the proved property : "$\frac{d}{dt}\boldsymbol{w}(t) = -H(t)\boldsymbol{w}(t)$, satisfies the constraint $\|\boldsymbol{w}(t)\|_2 = e^{-\lambda_0 t}\|\boldsymbol{w}(0)\|_2$, implying that $\boldsymbol{w}(t) \to 0$", we have $\|\text{vec}(W(t))\|_2 = e^{-\lambda_0 t}\|\text{vec}(W(0))\|_2 \to 0$ and there is no chance for W to learn any meaningful features.

5.4 In the case when both the Stop-Grad **and** the predictor are removed($W_P = I$), the gradient of $W(t)$ is :

$$\dot{W}(t) = -\frac{\partial J}{\partial W(t)} = -W_P^\top (W_p W(X + X') + WX) = -WX'$$

$X'$ is positive definite matrix with the same property as the previous question $W(t) \to 0$ .

5.5 We can see from Q5.2–5.4 that by combining the stop-gradient and the predictor, the $\dot{W}(t)$ won't become zero as time passes. As a result, the $W$ can learn some useful features in addition to the training.