GEM:

$$\min \; g^T G^T v + \frac{1}{2} v^T G G^T v \;\; s.t. \;\; v \geq 0$$

Variable	Shape
memories (G)	(t-1, p)
gradient (g)	p vector
$D = GG^T$	(t-1, t-1)
$d = -Gg$	t-1 vector
$A = E$	(t-1, t-1)
$m = 0$	t-1 vector

ModGEM:

$$\min \; g^T \tilde{G}^T v + \frac{1}{2} v^T \tilde{G} \tilde{G}^T v \;\; s.t. \;\; v \geq 0$$

Variable	Shape
memories (G)	(41*(t-1), p)
memory (G_i)	(41, p)
gradient (g)	p vector
relatedness (h)	41*(t-1) vector
relatedness (h_i)	(1, 41)
$\tilde{G}_i = h_i G_i$	(1, p)
$\tilde{G} = np.vstack(\tilde{G}_i)$	(t-1, p)
$D = \tilde{G}\tilde{G}^T$	(t-1, t-1)
$d = -\tilde{G}g$	t-1 vector
$A = E$	(t-1, t-1)

Question:

1. Generally:

   If we ignore the relatedness and only consider modularization here, is it possible to reconstruct the overall gradient in GEM from group gradients in ModGEM?

2. Specifically:

   Let's say the current task is task 2 now, so the previous task is task 1 only. What GEM outputs is an overall gradient whose shape is (1, p). What ModGEM outputs are 41 individual group gradients whose shape is (41, p). How can we combine 41 individual gradients so that the combination can be as close as possible to the overall gradient/the effect of the overall gradient in GEM can be represented by that of the combination of 41 individual groups?