![תמונה שמכילה טקסט, אוסף תמונות

התיאור נוצר באופן אוטומטי](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RD0RXhpZgAATU0AKgAAAAgABAE7AAIAAAAUAAAISodpAAQAAAABAAAIXpydAAEAAAAWAAAQ1uocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAANeQ15XXqNeZ16og15PXnteR15UAAAWQAwACAAAAFAAAEKyQBAACAAAAFAAAEMCSkQACAAAAAzM1AACSkgACAAAAAzM1AADqHAAHAAAIDAAACKAAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDIyOjA1OjE1IDE1OjA0OjU2ADIwMjI6MDU6MTUgMTU6MDQ6NTYAAADQBdUF6AXZBeoFIADTBd4F0QXVBQAA/+ELJmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjItMDUtMTVUMTU6MDQ6NTYuMzQ1PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPteQ15XXqNeZ16og15PXnteR15U8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgAywNGAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArntd8eeGvDWoCy1vVY7S5KCQRsrE7SSAeAfQ10NfNH7QyMPiJbMVIVrCPBx1+d63oU1UnysTPpSCaO4t454W3xyKHRh3BGQafXi+mftCeHrPSbS2m0rU/MhgSNiojIyFAOPm6cV1+p/FXSrDwhZeJrewvr7TLpijSQKuYWBxtcEjHIIz049xSlQqJ2sFzuaK5bwR8QdH8eWc0ulGSGa3bEttOAJFB6NwTkH1rF8W/GTRvB/iSXRtQ0+/lmiVWMkIQqQwBGMsD3qVSm5cttRnodFcT40+KOk+CP7P+32l5ci/iMsRtwvAGOuSP7wq54H+IWkePba5k0lZ4ZLUqJYbhQGAOcHgkEcGj2c+XmtoB1VFed6Z8ZtD1LxkvhsWN9b3LXLWwkmVAm8Ejs2eSMdO9N8WfGvw74W1iTTPJudQuYTtm+zhQsbf3ck8n6U/Y1L2sFz0aiuK8E/FTQfHF09nYie1vVUv9nuAAWUdSpBIP8AOubuv2hfDtpeTW76VqjNDIyEhY8Eg4/v0KjUbtYVz1miuA8IfGLw54v1dNMtkurO7kB8pLlVAkwM4BBPOO1QeKvjTonhLxFcaPfafqE01vt3PCqFTkA8ZYHvR7GpzcttR3PRqK878M/Gzwt4k1KKwBudPuJmCRC7QBXY9BuBIBPvV3xz8U9K8B6nb2WpWV5cSXEPnK1uFIAyRg5YelHsqnNy21A7eivIl/aM8NFwG0vVFBPJKR8f+P16ZoOu2HiTRYNV0mbzrWcZU4wQRwQR2INKdKcFeSA0aKhu7lbOynuZAWSGNpGC9SAM/wBK8n/4aN8N/wDQJ1X/AL5j/wDi6IU5z+FAevUV5RY/tC+Fbq8SG4tNRtEY4M0kaFV+u1if0ruPFXjPSfCHh9dX1OR3t5WVYVhG5pSQSMdugJzQ6U4tJrcDforyD/ho3w3/ANAnVf8AvmP/AOLq3pn7QPhbUNRitZrXULNZWCiaZEKqT64YnFX9Xqr7Iro9Uorh/HHxU0nwJqNtZ6jZ3ly9xD5ytbhCAMkd2HpXMf8ADRvhv/oE6r/3zH/8XSjRqSV0guev0VxngT4maZ4+uLyHTLO7t2tEVnNwFAO4kcYJ9K5ub9oPw3balJaT6dqa+VKY2k2IQMHBP3s0vY1G2rbBc9XornNQ8caTZ+BW8WQGS804Irr5IG5gWC4wSMEE8g+lcJ/w0b4b/wCgTqv/AHzH/wDF0Ro1JbIZ69XN6r8QfC+iau2l6pq0VveLt3RMjEjcMjoMdCK5rw58cfDHiHWIdN8u8sJZ2CRNcouxmPQZUnGfevIvjSzWvxhup2QkKsDgHjcAi/4VrSw7lPlnoK59TA5GRRXkEf7RnhvCB9K1RegJ2xkD/wAer1LR9WtNd0e11PTnL211GJI2IwcH1HY1jOlOHxIZdoqtqN6mm6ZdX0ys0dtE0rKvUhRk4/KvKv8Aho3w2P8AmE6r/wB8x/8AxdEKc5/CgPXqK8iX9ozw0WAbS9UUE8kpHx/4/XbP4/0ZvAc3iyyaS70+FcssagSA7gpXBIwQTTlRqR3QHT0V5r4b+OPhzxJr1tpMdrfWk104jiedF2lj0HyscZqa9+M+had40fw3eWV9HOl0LZp2VBGCSBu+9nbznp0o9jUvawrnolFcH41+LugeC9SGnXCXF7egBpIrYD90DyNxJ6+1R+DvjH4e8YaommwpcWN7Jnyo7gDEmBkgMCecdjij2VTl5raDPQKK878W/GbRfB/iSfRr/T7+eeAIWeFUKncoYdWB6GsT/ho3w3/0CdV/75j/APi6aoVGrpCuev0VxWg/E/S/EHg/VfEVrZ3kdtpe7zY5Au9tqhuMNjoe5rlv+GjfDf8A0CdV/wC+Y/8A4uhUajbSWwXPXqK8hX9ozw0WAbStVUE8nbHx/wCP12up+PbGx8Ew+KbOzu9R06RQ7G2Vd8an+JgxHAPB9KUqNSNroLnU0Vxfgf4o6H47uJ7bT1ntbqEbvIuQoZ17suCc471F43+Kul+A9WgsNTsL2d54RMjwBCuMkY5Yc8UvZT5uW2ozuaK4vX/ifpPh/wAI6X4intbue11Pb5SRBd65Utzkgdsda5X/AIaN8N/9AnVf++Y//i6qNGpJXSFc9eorh/BfxY8P+NtQawsBcWt4FLrDcoAXA64IJH4V3FZyjKLtJDCiuN8d/EvTPAM1nHqdnd3BvFdk+zhTjaRnOSPWuT/4aN8N/wDQJ1X/AL5j/wDi6uNGpJXSFc9dd1jRnc4VRkn0Fc7ovxA8MeItSFho2rRXV0ylhGqMCQOp5FUPCvxG0Tx5p96mkedHcwwlpLadQHAIxkYJBGfSvnL4deK7XwV42XVtQt554VikjKQgbst9SK1p4dyUk90Fz6/rm774heFtM1ptIvtXiivldY2hKMSGbGBkDHcVyug/Hjw1rmt22m/Zb6ze5cRxyzomzcTgA4YkZNeQ/FKc6f8AGy+u5I2Kw3EMu3puAVDx+VOlh3KXLPTQLn1ZRXkSftGeGmdQ+l6ogJ5O2M4/8frrfE/xK0fwz4Z07XZIri8s9RKiA24XOCu4EhiOwrF0aiaTW4XOworya2/aI8LTXCpPYanboesjRoQPwDZrtNa8daVpHgoeKIhJqGnHYVNqASQxxnkjGD1zQ6NSLSa3GdLRXF+GfiloHiXw9qOsKZrG200/6QLkDcoIyCApOc9B71ysn7Rfhxbooml6k8IbAlwgyPXG6hUajbSWwrnr1FYen+L9K1bwlL4i0yVriyiieRgBhxsGSpB6GvPf+GjfDf8A0CdV/wC+Y/8A4uiNKpK9lsM9eoryvTf2gvCt9fR29xbX9krsF86ZFKL9drEgV6mrB1DKQVIyCO9TOnKHxIBaKKKgCtf6jZ6Xa/aNRuYrWHeqeZKwUbmOAM+5qz16V83/AB78aHVfESeHbKT/AETTTmfaeHmI/wDZQcfUmtT4Q/F7yPI8OeKrj91wlpeyH7nojn09D26Gur6tL2fOhXPfKKAc9KK5RhRRRQAUUUUAFFFFABWdF4g0mbXpdFiv4W1KGMSPbBvmVf8APb3rgPiv8V4fCVs+k6JIkusyLhmHItQe5/2vQfiff5vttZ1G01pdWt7yZL9ZfNFxu+bd6k967KWFdSPM9Owrn3BRXA/DH4nWnjnTxbXZS31mBf30IOBKP76e3qO1d9XLKLg+WQwoooqQCiiigAooooAKbJIkUbSSsqIgLMzHAUDqSaJJEijaSVlREBZmY4CgdSTXzf8AFv4tv4ikl0Lw5KyaUh2zzqcG6I7D/Y/nWtKlKrKyEfRVhf2uqWMV7p86XFtMN0csZyrD2qxXgX7PvjMxXU/hW+k/dy5nsyx6N/Gg+o5/A+te+0Vabpz5RhRRRWQBRRRQAUUUUAFFFFABRRRQAVQ1PQtJ1nZ/a2m2t6Y/uG4hVyv0yOKv1Q1vWrPw9otxqupuyWtsAZGVSxAJA6D3Ipq99APGfj74b0XR/DOmXGk6XaWUrXZRngiCFl2E4OPcVqeAdS0fSv2dzceJIxNp5M6SQkZMpMhAUe5P5de1cJ8WfiLB8QLyw0vw9bzva28hZSyfPPIeBhRzgD8eau/EbRr3wp8GvC2i3AZHaeSa5XPSQgsFP03H8q9JQbpwhPdsk4LQNev/AAj4hg8Q6NDLDbec6xrISVlQEboy3fgjP4Gui8TXDfFj4rRnw7BKUuUhj+Zf9WoUb2b2BJ59q9C8C+DbHxn8AYtMl2rO000sM2OYpgxAP0xgH2rhvhZ4pk+Hnjy40nXYlgguZPs10zr80Lg4DZ67c9e2DntWvOpOUor3loB7L8VND02X4YalJcWcU81hZ4tppEBePGOh7dK4D9mz/j81/wD65w/zevctQ0+01fTZrHUIVuLW4XbJGx4cfhWfoPhHQfDDzPoOmxWTTgCUxkncBnHUn1NefGqlScH1H1Pnv416HL4X+JCavp7eUt9i7iZTykqkbv1wfxr0n4XfD/RbvwNFqniDToNRv9X3TzS3KbjhicAZ6cc5HPNeefHi+k1b4mW+mQfObWCOFVB/jc7v/ZgPwr1nWviH4f8AhtHp+g6ml00kNlHs8iIEFQNo6kc/Ka6ZubpQit2LqeI/DqFbD46Wltalkiivp4lGf4QHAH5Cvo+TwR4Xldnk8P6azOSWY2yZJPU9K+cPhnK2q/G+zvLaJ9kt3NPgjlVIc8/nX1ZUYttTXoNHyL4ZjWx+NNnDaDyo4dYMaKP4VEhAH5V9SX3hTQNTu3utR0axurh8bpZbdWY9upFfLuh/8lwg/wCw2f8A0aa988SfGDw34W16fSdTW8NzBt3mKIFeQCOc+9XiYzlKPLvYSPEvjP4csvC3j5F0WAWcE9uk6xxnARskHHpyM19EWek6R4p0HStS1nTLS+mlsonDzwq5G5Q2Bkepr5s8d+In+J3xChfRLSULIEtbWJ/vNyeTjpyT9BXvWu/EDQvhvBpmjawLppFs02GGMMNqjZ6j+7SrRm4Qj9oEee/H7wno+j6TpOoaRp1vZSNO0Mn2dAgcbdwyB6YPPvXU/s+NIfhzKHzsF6+zP+6ua84+L/xM0rxzaadZ6LDcLHbSNLJJOoXJIwAACfevT/g14j8PXPh+Lw7oH2hpbCAS3EksW0O7H5iOfU/liioprDpSWodT0W8khisZ5LoZgSNmkBGcqBzx34rznwxqfww8XasdO0TRbKW5WIylX04INoIB5I9xXfa3/wAi/qP/AF6y/wDoBr5z/Z7/AOSjzf8AXhJ/6ElYUoJ05SvsMX49aBpWheKdP/sexhsluLXfIkK7VLBiM46Dit34qEt8CvBpYkkpb5J/69zVP9o//kaNI/68z/6Ga3fH+h3er/s+eHZrGNpWsLa2nkRRk7PJ2k/hkH6ZrqjL3abYi18E/Cegal8O0vdR0izu7mW5kDSTxBzgHAHPQV5r8a9H0/RPiK9vpNpHaQvbRyGOJdq7jkEgduldl8GviZ4f0Dwm+i69dfYpYZnljkZCVdWwccA4IOa4nxpqn/Cz/iqo0CGSSOYx20GV5ZR1cjsOSfpVU1NV5OWwdD6Vfw7o2s2dnLq2l2l7IkCqrzwq5Ax05rxD4/6dpui3GjWekaZZ2SSpJLI0ECozkEAAkDp1/OvoWCLyLeOIHPloFz9BivAf2kv+QzoX/XvL/wChCuXCtuqkN7Hp3wq0fTtP+Huj3VlZQw3F1ZxvPKiANKevzHvyTXn/AO0B4d0fTPD+nXunabbWtzLdsskkMYUuCpJzjrzXpHgC4S0+E+i3MudkOmpI2BzgLk14z8XviZonjbQrGy0ZLoSQXBlczRhQBtI9T61VFTde67h0O1+Gl/pGm/AUXXiSNZtNjmk85Hi8wEGTA+XvyRW54UPw48afaf7B0Oxl+zbfM8ywCY3Zx1HPQ1wemf8AJqd9/wBdW/8AR61Y/Zs/1evfWH/2aqnD3ZzvrcDg/Hel2Wi/Gqey0uBba2ju4GSJOAu5UY49OSa+oNS8PaNrEqSatpVneSINqvPArkD0yR0r5q+J/wDyXm7/AOvm2/8ARcdfSPiPxJp3hTR21TWZHjtVdULIhY5JwOBSr8zjTtvb/IEeG/tAaBpOitoraRp1tZGUSh/s8YTdjbjOPqa9Y+FH/JK9B/69/wD2Y14R8TfGh+JnimxtPD1ncPBADFboV+eZ2PJ2jp0FfRvg/RW8O+DtL0mUhpLW3VJCOhbq36k0VrxoxjLcFua0sUc8LxTIskcilXRhkMD1BFfLHiC2tJvj0NPFjbRWUepxW4toogqFAyjBA655z9a+lvEniCz8L+H7nWNSEhtrbbvES5b5mCjA+pFfMWmXx8YfHS11DToJFS71VJ1RhllQMCScegBNGFTtKXSwM9T+NPhDw9p/w7mvtP0eztLmGeMJLBEEIBbBHHUVW/Z9tYNQ8C6xZ30KXFs94N0Mq7lb5F6g/QV0Xx1/5JXef9d4f/Qqwv2cf+RS1X/r9H/oApJt4Zt9w6nmNvawWPx/gtrOJYYIteRI40GAoEwwBXTftCeGzp/iSz8Q2w2pfJ5cuOMSoOD+K4/75r2f/hX/AIWOtjWP7Gg/tAT/AGjz8tnzM7t3XGc8149+0dqvm67pGko3EEDTuvu7YH6KfzrWnV9pVjy9hdDovg94PsNf8P3HinxRaR6pqGpTOA92gcBF+XIB4ySDz7CvLp7ODQvjulrpaeRBba0ixIDnaPMHH0r2mDxjovwt8FeHNM1pLjzJrIMPIjDc4BbPI7tXidvfp4q+N1vf6bHII73V0ljVx8wXeDk49hmqpOTlOT2A+oL/AMK6Bql411qOjWN1cOAGlmgVmOBgckelfOXxvs7LT/iBDp+m2FtZW0VrGQlvEE3FiSScdT0H4V9R18w/Hn/kqh/69Yf61hhG3UsNnuxsdI8M/Dy8ltdItjaxWDTzWqIFWfbHkhuOcgYyc1wXw68QeE/H+tXOnjwJptj5EBm8zakm75gMY2D1r0DxN/ySvVf+wPL/AOiTXh3wDvoNM8Ra3f3ZK29rpbzSkDJCqyk8d+BSpx5qcpdQNb9oDw7o+jWOjS6TplrZPJJIrm3iCbgAMZxXefDa9sdN+B9leasyLZQ28rT7xkFdzZGO+emO+a8j+KfjhPiV4g03TvDVrcSw25ZIgU+eeRyOQvYYA/Wu4+Iek3PhT9nmw0dm2yo8KXOw8Ekl2H03fyrWUW6cKct2xdTxltTk03xPL4i8J29xYWcF3m2LHcI85IRj05APHp610fj7xOfij4j0IaNZyNeNaJBJAAf9cWJIH+zyDmu3+Cvh3TvE/wALNe0vUUDLc3pVj/EhEalGHuCSa4fwzqV78IvijJBq0KvHGxt7k7MlomIIkQ9emG9+ldHMnJ2XvRA9w1dvC3gjwHoth4zgiuba3VII99v5w80JyQMcd6saBovgLxdoS6jpOgafJZzFkDNZqhODg9siuO/aGniuvAujXFtIssMt4HR1OQymNiCK6D4Ff8kpsv8ArvN/6Ga4XFqj7S7vcfU8Z+G8CWfx0tLa3ysUV7PGoz/CA4Ar6rr5Y8A/8l+g/wCwhcfyevqenjPjXoCOK8feIPBeizWS+NbSK4eVXNv5lp52ACN3bjqKr3fhrwXrvgC61TTdCsfs9xYyTQyC1EbjCkg9Mg5FcB+0n/x/aB/1zm/mld/4R/5INaf9geT/ANAaly8tKM09w6nkP7PrEfEiQA4BspAR68rX0BN4M8M3M7zT6BpsksjbndrVCWPqeK+fv2fv+Skv/wBeUn81r23xp8S9C8Dyrb6r9oe6kh86KGGPO8ZI+90HINXiVJ1rRBbHgHiLT7TS/jz9j0+BLe2j1WDZEgwq5KHgfU19N6j4b0TV7gT6ppNleTAbRJPArtj0yRXzZ4I0/UPiL8XhrMluVgW8+23Lj7sSg5Vc+vAAr6P8TeKNM8I6T/aWtSPHbeYI8pGXO45xwPpRib3jFb2BHg/x+0HStF1bRzpGn29l50L+YIIwgbDDGQPrXrfgfR9O1n4W+HYdWsbe9jS0RlSeMOFOMZANeEfEDxRcfFTxxaQeH7KZ4418i1jI+d8nJY+n9AK93vvEGlfCvwTo1vrHnPHHGlqDAm4lwmSevTg06qkqcIfaA4/42eB9AsvAb6tpemW1ldWs0Y328YTcrHaQQOD1B/CqXwWtB4o+FWv+Hrx/3LSskZP8G9AQfwYZrI+Knxf0nxb4U/sXQoLrM0yvNLOgUBV5wBk5OcV1vwW0+Xwv8Lb/AFnUIzGLhnu1VxgmNE4P44OKb540LS3voLqeKeHNJ1C58Xjwg87wR3t4lvexoeG8tzn8sEivZvjD4H8OaT8L5bnTNJt7SeykiEUsSAMQXCkMercHvXDfAyyfWfipNqc43fZoZbhif77nb/7MT+FbvxQ+Lnh/xP4LvNE0yK8NzLKnzSxhVUK4Y9/atanO60VHpa4dC/8AAO5gg+H/AIhk1D57OGYvKhXcNnl/Nx34HSuk8LX3wy8Y6jLZaHotlLPFEZWD6eEAXIHUj1Irkfg5DInwh8WSuhWORZQjEcNiHn+dY/7On/I8aj/14H/0YtZ1IJupK+wGZ8c9C0zQvG8CaRZxWcc1qsjxxLtXduYZA7dBX0h4bJbwrpRYkk2cWSf9wV8//tE/8j1Zf9eK/wDobV7/AOGv+RU0n/rzh/8AQBWdZt0YNjW5p1ieMfEMfhbwhqGrykZt4j5YP8Uh4UfmRW3Xhv7RviDZa6XoEL/6xjdTAHsPlQfqx/CuejDnmogzwi5uJbu6luLhy8srl3Y9WJOSajpKK90k9y+EPxe8jyPDniq4/dcJaXsh+56I59PQ9ule+A5GRXwlXuHwh+L32fyPDniq4/dcJaXsh+56I59PQ9ulefiMN9uA0z32igHIyKK80oKKKKACvL/iv8V4fCVs+k6JIkusyLhmHItQe5/2vQfifc+K/wAV4fCVu+k6JIkusyLhmHItQe5/2vQfiff5muLia7uJLi5kaWaRizu5yWJ6kmu7D4fm9+ewmwuLia7uZLi5kaWaVizu5yWJ6kmoq1/DfhnU/Fesx6bo1uZpnOWbosa92Y9gK+mdC+D/AIb0vwbNol5breSXag3N2ygSFuxQ/wAIHb9c121a8KWjJsfLOnajd6TqMN9p0729zA4eORDgqa+pfhj8TrTxzp4trspb6zAv76EHAlH99Pb1Havnzx94B1HwJrRt7oGaylJNtdhflkHofRh3Fc5p2o3ek6hDfadO9vcwOHjkQ4KmlUpwrwug2PuWiuA+GPxOtPHOni2uylvrMC/voQcCUf309vUdq7+vHlFwfLIsKKKKkApskiRRtJKyoiAszMcBQOpJpXdY0Z3YKqjJYnAA9a+cPi/8WG1+aTQPDk5XSkO24nQ4Ny3oD/c/n9K1pUpVZWQhvxb+Lb+I5JdC8OSsmkods86nBuiOw/2P515HRRXtQhGnHliSXNJ1O40bV7XUbJylxayrKh9wc/lX2noerQa9oNlqlr/qruFZVHpkcj8DkfhXxBX0f+zz4gN94UvNGmfMmnzb4wT/AMs35/Rg35iuXGQvDm7DR6/RRRXlFBRRRQAUUUUAFFFFABRRRQAVW1HTbPV9PlsdSt0ubWYASRP0YZz/ADFWaKNgMXSPBvhzQZvO0fRrO0l/56JENw+hPIqxrXh7SfEdtHb65YQ3sUb70SUZCtjGf1rSoquaV73AoaPoem+H7E2ejWcdnb7y/lx5xk9TWbqngLwtrWoSX2q6Ja3N1JjfK6nLYGB+ldDRRzSTumBHb28VrbR29uuyKJQiLnOABgCpKKKkDAuPA3hm71o6tc6NbS35kWU3DAlt4xg9e2BUms+DPDviG8W61vSbe9nRNiySgkhc5x+prboqueXcDI0XwnoPh13fRNJtbJ5BhnijwxHpnrj2rXoopNtu7A52LwB4Vg1UalFodql4svnCYKdwfOd3Xrml1TwH4X1rUJL7VdEtbq6kxvlkUktgYHeuhop88r3uBjaN4Q8P+H5Wl0XR7SzlYYMkcY3Y9M9ah8QeB/Dvim7iude01LuaJPLRmdlwuc44I7k1v0Uc0r3vqBxX/Cn/AAL/ANAGP/v7J/8AFVr+H/BHh7wrcTT6DpyWksyBJGV2bcM5xyTW9RTdSbVmwGyxJPC8Uqh45FKsp7g8EViaN4K8OeH743mi6RbWdwUKGSJSDtPUfoK3aKlNpWQGNrfhDQPEdxHPrmlW97LEuxGlBJUZzitO1s7ezsYrO1iWO2hjEUcQHCqBgD6YqaijmbVgOM1D4S+CtSumuJ9Eijkc5byXaME/QHFa/h/wX4e8LbjoWlw2sjjDSgFnYem45NblFU6k2rNgFY+t+EtB8SSxSa7pcF88IKxtKCdoPUCtiipTad0BVt9Ns7TS0022t0js0i8pYVHyhMY2/TFc7/wq3wT/ANC3Zf8AfJ/xrrKKalJbMDLPhrRj4f8A7D/s6AaXjb9lC4TGc/z5puieF9E8N+d/YWmw2Pn48zygRux0/nWtRRzSta4GBf8AgbwzqmrNqmoaNbT3zMrNO4O4lQAD17ACtLVtG0/XtPax1i0ju7ZmDGKQZBI6GrtFHNLuBkaP4U0Hw+xbRdJtLNzwXiiAYj03da16KKTbbuwKmp6XZazp0thqlsl1azY8yKQcNggj9QKo6N4Q8P8Ah6ZpdE0i1s5XGGkjj+bHpnritmijmaVrgUtW0fT9d09rHV7SO7tXIZopBkEjkVFonh3SPDlvJBodhDZRStvdIhgMcYzWlRRd2sAVg6r4I8Na5qJv9X0e2u7ogL5sgJOB0HWt6ihNrYDH1rwloPiIwHW9Lt7026lYvNH3AcZA/IVHo/gvw3oF0bnR9GtLScjHmxx/MB7E9K3KKfNK1rgFYOr+CPDevah9u1jR7a7udoXzZFJOB0Fb1FJNrVAYPjRFj+HevIg2qumTgAdh5TV4V+z7bQ3vinWLW6jWWCfTWjkjbo6llBB/Cvo66tYL6zmtbuNZYJ0MckbdGUjBB/CsnRfB3h7w5dPc6HpVvZTSJsZ4gclc5x+lbwqqNOUe4iXR/Cug6AxbRtJtLNyMF4ogGI/3utWdW0bT9d09rHWLSO7tmYMYpBkEjoau0VhzO97jMvRPDWj+G4pY9C0+GySZg0ixAgMR0NV9Z8GeHPEN4t3rWkW15cKmwSSLzt64/WtyinzSve4GJdeDfD19o1tpN3pUE1haHMEDglYzz059zV7SdH0/QtPWx0i1jtLVCWWKMcAk5NXaKOZtWbAwLPwL4ZsNYGq2ejW0N8rtIJ1U7gxzk9fc1v0UUm29wMjW/CmheJHhbXdMgvmgBEZlBO0Hrj8hVy20qxtNJXS7a2jjsVjMQgUfKEPBH05q3RRzO1gMLR/BPhvw/ffbNG0e2s7jYU8yIEHaeo/Sn614P8P+IrqO51vSre9miTYjyqSVXOcfmTW1RT5pXvcCtYabZaVai20y0gtIB0jgjCL+QqHWNE03xBY/Y9Zs47y33B/LkGRkdDV+ild3uBlaP4Y0Pw+D/YulWtkW4LRRAMR6FutP1vw7pPiO2jt9csIb2KJ96JKMhWxjP5GtKijmd73A5i2+G3g20uFmg8OWAkU5BaLcB+B4roLuxtr6wlsruFZbaVDG8RHysvp9KnopuUnuwMfRPCeheG5Jn0LS4LF5wBI0QwWA6fzrMb4YeCncs3hyyLE5JKnk/nXV0Uc8r3uBTh0nT7fSjplvZww2RQxm3jQKm09RgetUNF8HeHvDt09zomk29lNImxniByVznH6Vt0UuZ9wMTWvBnh3xFdrda3pNvezomxXlBJC9cfrWxBBHbW8cECBIolCIg6KAMAU+ii7aswCvkn4waudX+KGqsGzHauLWP2CDB/8AHt1fWc0qwQSSv92NSx+gGa+H9Uu2vtXu7tzuaeZ5CfXLE/1ruwUfechMrwxPPMkUY3O7BVA7k19F+Mfgpa6n4NsP7DjSDWdPtEjPQC62ryG/2s5wfwPt4z8NtNGrfEjRLVl3IbpZGHsnzH/0GvsatMVVlCUeUSPhe6tZ7K6ltruJ4Z4mKSRuuGUjqCKir6h+Kvwqg8YWr6po6JDrUS/RbkD+Fv8Aa9D+B9vmO6tZ7K6ltruJ4Z4mKSRuuGUjqCK6aNaNWN1uJntfwh+L32fyPDniq4/dcJaXsh+56I59PQ9uhr30HIyK+Eq9w+EPxe+z+R4c8VXH7nhLS9kP3PRHPp6Ht0rlxGHv78BpnvteX/Ff4rw+E7d9I0SRJdZkXDMORag9z/teg/E+6fFf4sQ+E7Z9J0ORJtZlX5nHK2oPc/7XoPxPv8z3FxNd3Ek9zI0s0jFndzksT1JNZ4fD83vz2G2FxcTXdxJcXMjSzSMWd3OSxPUk1s+EfCGqeM9cTTtJiz3lmYfJCv8AeY/071Y8EeBtU8c6wLTTk2W8ZBuLph8kS/1PoO9fVvhTwnpfg7RU03R4dqD5pZW5eZv7zH/OK6q+IVJWW4kiDwX4J0vwRoq2WmR7pWANxcsPnmb1PoPQdq6KiivIlJyd2UZniDw/p3ifRZtL1iATW8o/4Eh7Mp7EV8oePvAOo+BNaNvdAzWUpJtbsL8sg9D6MO4r7CrM8QeH9O8T6LNpesQCa3lH/AkPZlPYiuihXdJ+Qmj4t0/ULvStQhvtOne3uYHDxyIcFTX1N8MfidaeOdPFtdlLfWYF/fQg4Eo/vp7eo7V8+eP/AADqPgTWjb3QM1lKSbW7C/LIPQ+jDuK5zT9Qu9K1CG+06d7e5gcPHIhwVNejUpwrwuidj7lpHdY0Z3YKqjJYnAA9a8/+GfxRsvGumGC/aO11e2TdPEThZFHWRfb1Hb6V5n8XPi42uPNoHhqYrpqnbcXKHBuT6D/Y/n9K82OHnKfIVcX4ufFxtcebQPDUxXTVO24uUODcn+6P9j+f06+O0VZ0/T7vVdQhsdOge4uZ3CRxoMljXrwhGnGyJDT9Pu9V1CGx06B7i5ncJHGgyWNfQVh8FrTR/hhq0N2sd1rt1alzKBkRMvzBE/EYJ7/Suj+GHwwtPA+ni6vAlxrU6/vZgMiIH+BPb1Pf6V6AQGUg9Dwa86vim3aGyKSPhKvSvgPq5034lRWrNiPUIHgI9SBuH/oNcV4p07+yPFuq2AXatvdyIo9FDHH6YqXwbfnS/G2jXgOBFexFj/s7gD+hNehNc9NruiT7TooorwSwooooAKKKKACiiigAooooAK5nXviJ4Y8M6n/Z+takLa52B9hjZuD0PA9q6avmD4//APJTT/15xf1rehTVSfKxM9m/4XJ4G/6Da/8Afl/8KP8Ahcngb/oNr/35f/Cvkqiu76nT7sVz61/4XJ4G/wCg2v8A35f/AAo/4XJ4G/6Da/8Afl/8K+SqKPqdPuwufWv/AAuTwN/0G1/78v8A4Uf8Lk8Df9Btf+/L/wCFfJVFH1On3YXPrX/hcngb/oNr/wB+X/wq7a/FHwVdsqxeIbQM3QSEp/MCvj2ij6lDuwufdFpe2t/AJrG5huYj0khkDqfxFTV8R6N4g1bw9erdaNfz2cqnrG5Ab2I6Eexr6U+FXxTTxvbtp+qCODWIE3EJws692UdiO4/yOWthZU1zLVDuekUUUVyDOKm+L3gm3uJIZtZVZI2KMPJfgg4Pao/+FyeBv+g2v/fl/wDCvlnXP+Rh1H/r6l/9DNUK9VYOnbdk3PrX/hcngb/oNr/35f8Awo/4XJ4G/wCg2v8A35f/AAr5Koo+p0+7C59a/wDC5PA3/QbX/vy/+FH/AAuTwN/0G1/78v8A4V8lUUfU6fdhc+tf+FyeBv8AoNr/AN+X/wAKP+FyeBv+g2v/AH5f/Cvkqij6nT7sLn1r/wALk8Df9Btf+/L/AOFH/C5PA3/QbX/vy/8AhXyVRR9Tp92Fz7a8P+JNL8Uae19odyLm3WQxlwpXDAA45+orUryv9nv/AJJzN/1/Sf8AoK16pXnVIqE3FFHI6n8UvCGjapPp+o6sIbq3fZInlOdp9MgVU/4XJ4G/6Da/9+X/AMK+d/il/wAlS1//AK+j/IVyVd8cJBxTuybn1r/wuTwN/wBBtf8Avy/+FKPjH4GLAf22oyephf8Awr5JoqvqdPuwufaWkeNfDWvSiLSdas7mU9IxJhz9FOCa3K+E0do2DIxVlOQQcEGvoL4I/Ey71i4PhnxBcNPcKhazuJDlnA6ox7kDkH0BrCtheSPNFjTPaqyvEPibSvC1gl7rl0LaB5BGrlS2WIJxx7A1q1j+KvDFh4u8Pz6TqiZjkGUcfeicdGHuP/rVxxtf3thnPf8AC5PA3/QbX/vy/wDhR/wuTwN/0G1/78v/AIV8zeLfCmo+DvEE2l6onzIcxSgfLMnZh/ng8Vh16SwlNq6bJufWv/C5PA3/AEG1/wC/L/4Uf8Lk8Df9Btf+/L/4V8lUU/qdPuwufbmheINL8S6aL/RLtLq3LFCy5G1h2IPINaVfIXw48fXXgXxCs4LS6dOQl3bg/eX+8P8AaHb8q+tdPv7XVNPgvrCZZ7a4QPHIp4YGuKvRdKXkNMsU2WVYYXlkOERSzHGcAU6kdQ6MrdGGDXOM4f8A4XJ4G/6DS/8Afl/8KP8Ahcngb/oNr/35f/CvljWrYWWvX9qowIbmSMD6MRVGvV+p031ZNz61/wCFyeBv+g2v/fl/8KP+FyeBv+g2v/fl/wDCvkqij6nT7sLn3FpOq2WuaXBqOlzie1nXdHIBjIzj+lXK8t/Z+vzdfDh7djn7HeSRgegIV/5sa9SrzakeSbiUcnq/xN8J6Dqs2napqqwXUBAkj8pztyM9QPQ1S/4XJ4G/6Da/9+X/AMK8C+Mf/JVtY/30/wDQFrh69CGEhKKbbJufWv8AwuTwN/0G1/78v/hR/wALk8Df9Btf+/L/AOFfJVFV9Tp92Fz61/4XJ4G/6Da/9+X/AMKP+FyeBv8AoNr/AN+X/wAK+SqKPqdPuwufX9v8VfBNyVCeIbVS3QSbk/mK6ax1Kx1OHzdNvLe7j/vQShx+lfDVXNN1bUNHvFutKvJ7SdDkPC5U/p1qZYKP2WFz7ioryX4TfF1vFMy6H4iKJqgUmCdRtFwAOQR2bHPHWvWq8+pTlTlyyKCuc8Q+PvDnhW+js9d1AW08kfmKpjZsrkjPA9jXR183/tF/8jxYf9eC/wDobVpQpqpPlYmer/8AC5PA3/QbX/vy/wDhR/wuTwN/0G1/78v/AIV8lUV3fU6fdiufWv8AwuTwN/0G1/78v/hR/wALk8Df9Btf+/L/AOFfJVFH1On3YXPrX/hcngb/AKDa/wDfl/8ACp4Pi34HuPu6/AnP/LRHX+Yr5Doo+pU+7C59wabrel6zH5mlahbXi4yfIlV8fUDpV6vhmzvrrT7lbixuZbaZDlZInKsPxFe6fC/41zXt5DofjCRWeUhLe/PGW7LJ25/vfn61z1cJKKvHUdz3KiiiuIZl+J5/svhPVp/+ednKwx/uGviY9a+zvHXmf8K/13yc7/sMuMf7pr4xr08F8LJZ6T8BbcT/ABSt3P8AywtZpB/3zt/9mr6kr5m/Z5/5KRcf9g6X/wBDjr6ZrDGfxRoK8z+Kvwqg8YWr6po6LDrUS/RbkD+Fv9r0P4H29MormhOUJc0RnwvdWs9ldS213E8M8TFJI3XDKR1BFQ19QfFj4Vx+L7RtV0WNI9ZhXkdBdKP4T/teh/A+3zJcW81pcyW9zE0U0TFXjcYKkdQRXs0aqqxuiBju0jlpGLMepY5Jrrfh/wDDzU/HerCO3DW+nxN/pN4y/Kg/uj1Y+n51yFegfC74mXHgbUvs15vn0a5cGaIcmJunmL7+o7iqqc6g+TcD6a8O+HdN8LaNDpmj26wwRjk/xSN3Zj3JrUqCyvbbUbGG8sZknt5kDxyIchge9T14Tbb1LCiiikAUUUUAZniDw/p3ifRZtL1iATW8w/4Eh7Mp7EetfKHj/wAAaj4E1o290DNZSkm1uwvyyD0Pow7ivsKvPfjH4i8P6V4NnsNdgS9uLxSLW0zht3aTPVQD3/CurDVZRlyrVMTPlVJHjbdG7IcEZU44PBFNoqzp2n3erahDY6dA9xczuEjjQZLGvYJDT9Pu9V1CGx06B7i5ncJHGgyWNfU3ww+GNp4G08XV4EuNanT99MORED/Ant6nvR8MfhjaeBtPFzeBLjWZ1/fTAZEQ/uJ7ep7139eViMRz+7HYpIKKKK4hnyR8YLX7L8VdZGMCSRZP++kBrjIXMc6OpwVYEH0rvvjjt/4WtqGzP+rizn18ta8+X7wr3qWtOPoQfdFrMLizhmHSSNXGPcZqWqmk/wDIFsf+veP/ANBFW68J7lhRRRSAKKKKACiiigAooooAK+YPj/8A8lNP/XnF/Wvp+vmD4/8A/JTj/wBecX/s1dmD/iCZ5hRS0V6xIlFLRQAlFLSUAFFFFABWp4b1yfw34ksdWtSQ9rMrkA/eX+JfxGR+NZdLQ0mrMD7otriO7tIbmA7o5kWRD6gjIqWuV+GV4198MtCmkOWFqIyf90lf6V1VfPyXLJos+INc/wCRh1H/AK+pf/QzVCr+uf8AIw6j/wBfUv8A6Gao1762IEopaKYCUUtJQAUUUUAFFFLQB9M/s9/8k5m/6/pP/QVr1SvK/wBnv/knM3/X9J/6CteqV4df+LIpHx98Uv8AkqWv/wDX0f5CuSrrfil/yVLX/wDr6P8AIVyVezT+BehIUUUVYBW34O1F9J8a6PexHDQ3kZP03AH9CaxKu6P/AMhyx/6+I/8A0IUmrqwH3DRRRXzxZyvj/wACWPjrw+1pcbYryIFrW5xzG3ofVT3H+FfJWs6NfaBq9xpmqQNBdW77XU/oR6g9Qa+364D4p/DaDxvo5uLJUi1m1Q+RJ080f882P8j2P1NdmGr8j5ZbCaPlCiprq1nsruW2u4nhnhcpJG4wVYcEEVDXrEhXrvwT+JH9g6ivh3WZsabdv/o8jniCU9v91v0PPrXkVLnHSoqQVSPKwPu2ivK/gt8Rf+Em0caJq02dUsU+R2PNxEOh+o6H8D616pXh1IOEuVlnxl49hEHxB12MYwt9L0GP4jXPV0vxE/5KRr3/AF/S/wDoRrmq92HwogKKKKoD6G/Zukz4d1qPJyt0jY9Mp/8AWr2ivGP2bodvhzWpsH57tFz9Ez/7NXs9eLif4rKWx8k/GP8A5KtrH++n/oC1w9dx8Y/+Srax/vp/6AtcRXrUv4cfQkSilorQBKKKKACiiigCzYX1xpmo299ZyGOe3kWSNx2YHIr7V0DVo9d8O2GqQY2XcCS4HYkcj8DkV8Q19W/BC8a7+FdgHOTBJLEOewYkfzrhxsbxUho9Br5v/aL/AOR4sP8ArwX/ANDavpCvm/8AaL/5Hiw/68F/9DaubCfxRvY8hopaK9ckSilpKACiiigApQcHI60lFAH1p8IfFj+K/AVu93J5l7ZH7NOx6tgfKx+q4/EGu6r54/Zx1Uw+I9V0tmO24thMo7ZRsfyf9K+h68TEQ5KjSKRl+J4PtXhPVoOf3lnKox/uGviY9a+6polngeJ/uyKVP0IxXw/qlq1jq93aOu1oJnjIPbDEf0rrwT0khM7/AOAtwIfilboes9rNGP8Avnd/7LX1JXx18NtSGk/EjRLp22J9qWNz7P8AKf8A0KvsWssYvfT8hoKKKK4hhXl/xX+FEPi22fVtEjWLWo1yyjgXQHY/7XofwPt6hRVwnKEuaIHwtcW81pcyW9zE0U0TFXjcYKkdQRUVfT3xX+FEPi22fVtEjWLWo1yyjgXQHY/7XofwPt8zXFvNaXMlvcxPFNExV43GCpHUEV7NKrGrG6IPRPhT8UZvBl6NO1Vnm0Wd/mHU27H+NR6eo/Hr1+obe4hu7aO4tZUmhlUOkiHKsD0INfCtep/CT4qyeFLpdH1yRpNHmb5HJybVj3H+ye4/H1zz4nD83vx3GmfTdFNjkSaJJYXV43UMrKchgehBp1eWUFFFcx478c6d4F0M3l6RLcyZW2tQcNK39FHc04xcnZAR+PvH2neBNFNxdETXsoItbQNgyH1Poo7mvk3Xtdv/ABJrU+qatO01zO2ST0UdlA7AdhUniTxHqPirXJtU1ebzZ5TwBwqL2VR2Aqpp2nXerajDY6dA9xczsEjjQZLGvZo0VSjd7kN3DTtOu9W1GGx06B7i5ncJHGgyWNfUvwx+GNp4G08XN2EuNZnX99MBkRD+4nt6nvR8MfhjaeBtPFzdhLjWZ0/fTAZEQ/uJ7ep7139cWIxHP7sdikgoooriGFFFBIVSTwByaAPkj4wXX2r4q6yc5Ecix/8AfKAVxkKGSeNFGSzAAetaXinUf7X8W6rf7twuLuR1PqpY4/TFS+DbA6p420azUbvNvYgw/wBncCf0Br34+7BeSIPs61iFvaQwjpGioMewxUtFFeAWFFFFABRRRQAUUUUAFFFFABWLqvg7w7rl79r1fR7S8uNoXzJY8nA6Ctqimm1sBzH/AArbwb/0Len/APfkUf8ACtvBv/Qt6f8A9+RXT0VXtJ9wOY/4Vt4N/wChb0//AL8ij/hW3g3/AKFvT/8AvyK6eij2k+4HMf8ACtvBv/Qt6f8A9+RXHeP/AIL6BeeHbq88OWQ0/ULaNpUSJjsmwMlSp6E9iK9YpkwBgkBGQVOQfpVRqzi7pgfCtJTn/wBY31pte6QFFFFAH1z8H/8AklOi/wDXN/8A0Y1drXFfB/8A5JTov/XN/wD0Y1drXg1f4kvUs5qT4deEJpXll8O2Du7FmYxDJJ6mm/8ACtvBv/Qt6f8A9+RXT0UuefcDmP8AhW3g3/oW9P8A+/Io/wCFbeDf+hb0/wD78iunoo9pPuBwHirwz4C8J+GbzWL7w3p5S3TKoIgDI54VR9TXy3ql9/aWpTXYtoLVZGysFum1Ix2AFe5ftH6y6Wuj6NGxCyM9zKM9cfKv82rwKvUwsXycz6ksKKKK6xHffB3wpYeLfHP2XV4jNaW1u1w8WSA5BUAEjtls/hX0Qvw08GIoUeG7DA9Ys14x+zn/AMjtqP8A14H/ANDWvo+vKxU5KpZMpFHSdF03QrM2uj2UNnAWLmOFcAse/wClXqKK4229xnP3vgTwtqV9LeX+hWVxczNuklkiBZj6mvKvjb8O9A0bwrHreh2MdjNFcLHKkXCOrZ5x6g46epr3WvNfj1/yS2f/AK+of51vQnL2kVcTPluiiivaJCruj/8AIcsf+viP/wBCFUqu6P8A8hyx/wCviP8A9CFD2A+4aKKK+dLCiivO/i18SE8FaQLPTmVtYvEPlDr5CdPMI/kPX6VcIOcuVAcF+0FZ+GVvoLq2uUTxAcCe3iGd8eOGf+6w7dyPpXiVTXV1Pe3UlzdyvNPKxZ5JGyzE9yahr26cOSCje5AUUUVoBe0bV7zQdYttT02UxXNtIHRv6H1B6EV9g+DfFtl4x8LwavZkKWG2eLPMMg+8p/mPYivjCus8C+PL7wTcXv2cGW2vbd45Id2AH2nY49wT+RNc2Io+0jdboaZj+Jbz+0PFWqXedwmu5XB9QWOKy6U80ldKVlYQUUVb0vTrjV9WtdPs0Lz3UqxRqPUnFAH038CNLbTvhjDM6lWvriSfnuOFH6LXpFUtF0uHRNDstMtv9VaQrEpx1wMZ/GrteBUlzTcizB1DwP4Z1a+kvdS0SzubmXG+WSPLNxiq3/CtvBv/AELen/8AfkV09FHPJdQOY/4Vt4N/6FvT/wDvyKP+FbeDf+hb0/8A78iunoo9pPuB5x4r+CvhjWdJnGj2Mel34UtDLASFLdgy5xg+3NfLc0TwTvDKpWSNirKexBwRX3XXxR4rUL4y1pVGAL+cAen7xq9DB1JSumyWZFFFFd4gr6h+AP8AyTFf+vyX/wBlr5er6h+AP/JMV/6/Jf8A2WuTGfwvmNHptY+r+EtA166S41nSbW9mRNivMm4hc5x+tbFFeSm1qijmP+FbeDf+hb0//vyKP+FbeDf+hb0//vyK6eiq9pPuBzH/AArbwb/0Len/APfkVj+I/g34S1rS5YbLTotMu9v7q4tgV2t2yvQiu/opqpNO6YHw7qum3OjatdadfJsuLWVopF9wcVTr03496Ytj8SWuI02i9to5Tjuwyp/9BFeZV7dOXPBSICiiirA9F+Bc5h+K1io6TQzIef8AYLfzUV9U18mfBj/krGkfWT/0W1fWdeVjP4i9CkFfJPxh0g6R8UNVXbiO6cXSe4cZP/j26vravDv2jdAL2ul6/Cn+rY2sxA7H5kz+TfnU4SXLUt3Bng0MrwTxzRHa8bBlPoRX2v4b1ePXvDGnapEcrdW6SH2JHI/A5FfElfRn7PXiYXvhy78PzvmWwfzYQT1jc8j8Gz/31XVjIXhzdhI9ioooryigooooAK8w+K/woh8W2z6tokaxa1GvzKOFugOx/wBr0P4H29Poq4TlCXNED4WuLea0uZLe6ieKaJirxuMFSOoIqKvp/wCK3woh8XWz6tosaxa1EvzKOBdAdj/teh/A+3zLcW81pcyW91E8U0TFXjcYKkdQRXs0qsasbohnrHwh+LB8Oyx6D4imZtLkbEE7HP2Unsf9g/p9K+kFZXUMhDKwyCDkEV8JV638NfjO/hbSZtK8QpNeWsMZazZDl0IHEZz/AAnse306c2Iw3N70NxpnuHjbxpp3gjQXv9QYPM2Vt7YHDTP6D29T2r5M8T+J9S8W65LqerzeZK/CIPuxL2VR2Ap/ivxXqXjDXZdT1aUszcRxA/LCnZVHp/Os7TtOu9W1GGx06B7i5nYJHGgyWNbUKCpK73BsNO0671bUYbHToHuLmdwkcaDJY19S/DH4Y2ngbThc3YS41mdf304GREP7ie3qe9L8MfhjaeBtOFzdhLjWZ0/fTgZEQ/uJ7ep7131ceIxHP7sdhpBRRRXEMKKKKACub+IOtjw94B1fUAwWRbdo4ucfO/yr+pz+FdJXhP7RXiYbdP8ADdvJ82ftVyAenZAf/Hj+VbUYc9RITPCK9K+A+kf2l8SorllzHp8Dzk+hI2j/ANCrzSvo/wDZ58P/AGHwpeazMmJNQm2Rkj/lmnH6sW/IV6uIly0mStz1+iiivELCiiigAooooAKKKKACiiigArzDx18Z4vBPiiTRn0R7wxxpJ5ougmdwzjG0/wA69Pr5a+PP/JVLn/r2h/8AQa6cNCM52kJnZ/8ADSsH/QsSf+Bw/wDjdH/DSsH/AELEn/gcP/jdeB0V6P1Wj2Juz3z/AIaVg/6FiT/wOH/xuu3+HHxNT4hSX6R6U1h9jVCSZ/M37s/7Ix0r5Mr3T9mv/j41/wD3IP5vWFfD04U3KKGme9U2X/Uv/umnU2X/AFL/AO6a8wo+FX/1jfWm05/9Y31ptfREBRRRQB9c/B//AJJTov8A1zf/ANGNXa1xXwf/AOSU6L/1zf8A9GNXa14NX+JL1LCiiiswCiiigD5s/aJlZvHdkjH5UsV2j0y7V5JXrP7Q/wDyP9r/ANeKf+hNXk1e5Q/hRIe4UUUVsB6/+zn/AMjtqP8A14H/ANDWvo+vnD9nP/kdtR/68D/6GtfR9eRi/wCKUtgooorkGFea/Hr/AJJbP/19Q/zr0qvNfj1/yS2f/r6h/nWtH+JH1Ez5booor3SQq7o//Icsf+viP/0IVSq7o/8AyHLH/r4j/wDQhQ9gPuGiiivnSynq+qW2i6Pd6lfPst7WJpXPsB0+p6V8Z+J/EF34o8R3mr37Ey3Dkhc8IvRVHsBgV7p+0P4jNn4fstBgfD30nnTAf8806D8W5/4DXzrXq4OnaPO+pLCiiiu0QUV7h8OfhxZeLvg1erOqx3tzdvJa3BHMZRQo/AncCPevGtT0260fU7jT9QhaG5t3MciN2IrONSMpOK3QFSiiitACiiigAr3X4BeBGMreLdTiIVQY7BWHU9Gk/oPxrjfhd8L7rxtqC3t+rwaLbv8AvZOhnI/gT+p7fWvqa1tYLK0itrSJYYIUCRxoMBVHAArgxVdJckdxpEtFFFeYUFFFFABRRRQAV8UeLf8AkdNb/wCwhP8A+jGr7Xr4o8W/8jprf/YQn/8ARjV6GC3kSzIooor0hBX1D8Af+SYr/wBfkv8A7LXy9X1D8Af+SYr/ANfkv/stcmM/hfMaPTaKKK8goKKKKACiiigD54/aQTHiXR5P71ow/J//AK9eMV7R+0i4PiPRk7raOfzf/wCtXi9e3h/4SIe4UUUVuB6J8DITL8WNPbGRHFM54/6ZsP5kV9VV83/s66cZ/GmoX5GVtbMpn0Z2GP0U19IV5GMd6pSCsTxj4ej8U+EdQ0iUDNxEfLJ/hkHKn8wK26K5U2ndDPha5t5bS6ltrhDHLC5R0I5Ug4Ird8CeKJfB/jGy1ZCxiRtlwg/jibhh/Ue4Fd58e/Bh0rxCniKyjxaaicT4HCTAdf8AgQGfqDXkNe7GUasL9yNj7ptrmG8tIrm1kWWGZBJG6nhlIyDUteH/AAG+IAmtx4S1WbEkYLWDufvL1Mf4dR7Z9K9wrxatN05crLCiiiswCiiigArzD4rfCiHxdbPq2ixrFrUS8qOFugOx/wBr0P4H29Poq4TlCXNED4Wubaa0uZLe6ieGaJijxuMMpHUEVFX0/wDFb4Uw+LrZ9W0WNYtaiXkDhboDsf8Aa9D+B9vm+20XUrvWl0i3s5n1BpfKFvtwwbuCO1ezSrRqRuRYi07TrvVtRhsdOge4uZ2CRxoMljX1N8MfhjaeBtOFzdhLjWZ1/fTgZEQ/uJ7ep70fDL4ZWngbThcXQS41mdP304GREP7ie3qe9d9XBiMRz+7HYpIKKKK4hhRRRQAUUUUAVNV1O10bSbrUr+QR21rGZJGPoP69q+MfE2vXPibxJe6veE+ZdSlgufuL0VfwGBXq3x4+IC310PCukzboLdw166nh5B0T6Dqff6V4pXrYSlyx5nuyWXNJ0y41nWLXTbJC891KsSD3Jxn6V9p6FpEGg6DZaVaf6q0hWJT64HJ/E5P414j+z74MMt1P4qvov3ceYLPcOrfxuPoOPxPpXvtc2Mqc0uVdBoKKKK4hhRRRQAUUUUAFFFFABRRRQAV8tfHn/kqlz/17Q/8AoNfUtfLXx5/5Kpc/9e0P/oNdmD/ifITPNqKKK9YkK90/Zr/4+Nf/ANyD+b14XXun7Nf/AB8a/wD7kH83rnxP8Jgtz3qmy/6l/wDdNOpsv+pf/dNeKWfCr/6xvrTac/8ArG+tNr6IgKKKKAPrn4P/APJKdF/65v8A+jGrta4r4P8A/JKdF/65v/6Mau1rwav8SXqWFFFFZgFFFFAHzV+0P/yP9r/14p/6E1eTV6z+0P8A8j/a/wDXin/oTV5NXuUP4USHuFFFFbAev/s5/wDI7aj/ANeB/wDQ1r6Pr5w/Zz/5HbUf+vA/+hrX0fXkYv8AilLYKKKK5BhXmvx6/wCSWz/9fUP869KrzX49f8ktn/6+of51rR/iR9RM+W6KKK90kKu6P/yHLH/r4j/9CFUqu6P/AMhyx/6+I/8A0IUPYD7hooor50s+UfjZq/8AavxQv0Vsx2Spapz02jLf+PE15/Wr4ovG1DxZqt25y015K+fq5rKr36ceWCRAUUUVYH1r8GYPI+E+jjGNwkf65kY1y/x0+H39r6YfE2lRZvbNMXSKOZYh/F9V/l9K6/4S/wDJK9D/AOuJ/wDQ2rsWVXQq6hlYYIIyCK8V1HCs5LuV0PhKiu8+LXgU+DPFjG0jI0y+zLakDhP70f4E/kRXB17EZKcVJEhVzSprO31e1m1O2N1ZpKpnhV9pdM8gHtxVOiqA+3dAn0y58P2U2giJdOeFTbrEuFVfTHb/ABrRr5x+BnxC/sbVB4a1WbFjeP8A6M7niGU9vYN/P6mvo6vDrU3TnZlIKKKKxGFFFFABRRRQAV8UeLf+R01v/sIT/wDoxq+16+KPFv8AyOmt/wDYQn/9GNXoYLeRLMiiiivSEFfUPwB/5Jiv/X5L/wCy18vV9Q/AH/kmK/8AX5L/AOy1yYz+F8xo9NoorjPGPxR0PwRqkNhq8V48ssQlUwRhhjJHcj0ryoxcnaJR2dFeV/8ADQvhH/n31P8A78r/APFUf8NC+Ef+ffU/+/K//FVp7Cr/ACiueqUV5X/w0L4R/wCffU/+/K//ABVVr/8AaJ8ORWbtp2n6hcXGPkSRVRc+53E4+gp/V6v8oXRxX7RN9HP44srVDl7ayAf2LMT/ACxXkdaOva3eeI9dutW1Jw1xdPvbHRfQD2AwKzq9inHkgokhRRWjoI0o61bnxC866erbphboGdwP4RkjGeme1W9EB9F/APw62k+Bn1K4TbNqkvmLkc+WvC/mdx/EV6lXk9v8ffBdpbRW9tZ6jFDEgSNFgQBVAwAPmrr/AAX8QNJ8dpdto0dygtSofz0C53ZxjBPpXi1oVG3OSKOpooornGZfiTw/Z+KPD13pGormG4TG7HKN1Vh7g4NfHXiTw/e+F/EF1pOpptmt3xuxw69mHsRzX2zXn3xX+HMfjfRBcWKqmsWakwMePNXqYz/Q9j9a68NW9nLlezE0fLFrdT2V3FdWkrQzwuHjkQ4KsOQRX1Z8L/iPb+ONGEV0yRaxbKBcQjjzB08xfY9x2P4V8pXFvNaXMlvcxNFNExR43GCpHBBFWdH1i+0HVoNS0q4a3uoG3I6/yPqD6V6FaiqsfMlH3BRXBfDf4o6d44s1trgpaaxGv722JwJPVk9R7dR+td7XjSjKDtIsKKKKkAooooAKzovD+lQ69LrUVhCupTRiN7kL8xUf56+wrRop3aAKKKKQBRRRQAUUUUAFeZ/Fz4mx+ENNbS9JlVtauU42nP2ZD/Gff0H4/Vfid8W7Pwhby6bpDx3WtMMbQcrbe7ep/wBn86+Y729udRvpry+nee4mcvJLIcsxPc13YfD8z557CbInd5ZGkkYu7ElmY5JPrW74L8KXnjLxPbaVZgqrHdPLjiKMfeY/09yKydPsLrVdQgsdPhee5uHCRxoMlia+s/hr4AtvAnh4RNtk1K5Ae7nHc9kH+yP1612V6ypR8yUjptJ0u10TSbbTdOj8q2toxHGvsO59z1q5RRXi7lhRRRQAUUUUAFFFFABRRRQAUUUUAFfLXx5/5Kpc/wDXtD/6DX1LXy18ef8Akqlz/wBe0P8A6DXZg/4nyEzzaiiivWJCvdP2a/8Aj41//cg/m9eF17p+zX/x8a//ALkH83rnxP8ACYLc96psv+pf/dNOpsv+pf8A3TXilnwq/wDrG+tNpz/6xvrTa+iICiiigD65+D//ACSnRf8Arm//AKMau1rivg//AMkp0X/rm/8A6Mau1rwav8SXqWFFFFZgFFFFAHzV+0P/AMj/AGv/AF4p/wChNXk1es/tD/8AI/2v/Xin/oTV5NXuUP4USHuFFFFbAev/ALOf/I7aj/14H/0Na+j6+cP2c/8AkdtR/wCvA/8Aoa19H15GL/ilLYKKKK5BhXmvx6/5JbP/ANfUP869KrzX49f8ktn/AOvqH+da0f4kfUTPluiiivdJCruj/wDIcsf+viP/ANCFUqu6P/yHLH/r4j/9CFD2A+4aZMxW3kYdQpI/Kn0hGVI9RXzpZ8M3rF9QuGbqZGJ/OoKuavGYdbvojwUuJFP4Map19EtiAooooA+ufg/J5vwp0U4xiN1/J2FdrXAfBGbzfhLpmTko8yHjp+9b+mK7+vBq/wASXqWcx8QfCEPjTwhc6a4UXKjzbWQ/wSgcfgeh9jXx9dW01ldy211G0c0LlHRhyrA4Ir7or54+P3gr7DqkXiiwixBeER3YUcLLjhv+BAfmPeuvCVbPkfUTPGKKKK9MkVWKsCpIIOQR2r6o+D3j8eMPDn2PUJQdWsFCy56zJ0WT+h9/rXytWz4V8SXnhPxJa6vp7fPC3zoTxIh+8p9iKwr0lVjbqCPtWis/Qdbs/Eeh2uq6bJvt7lA6+qnup9weDWhXitNOzLCiiikAUUUUAFfFHi3/AJHTW/8AsIT/APoxq+16+KPFv/I6a3/2EJ//AEY1ehgt5EsyKKKK9IQV9Q/AH/kmK/8AX5L/AOy18vV9Q/AH/kmK/wDX5L/7LXJjP4XzGj02vm/9ov8A5Hiw/wCvBf8A0Nq+kK+b/wBov/keLD/rwX/0Nq48J/FG9jyGiiivXJCiiigAooooAKKKKACvfP2bP+PXXv8Afh/k1eB175+zZ/x669/vw/yaubFfwmNbnuVFFFeMUFFFFAHlnxY+E8Xiq3k1jQY1j1mNcvGOBdAdj/teh79DXzTc201ncyW91E8M0TFHjdcMpHUEV901wPxF+FOmeOIWuoNtlq6rhLkL8snorjv9eo9+ld2HxPJ7s9hNHypbXM1ncx3FrK8M0bBkkjYhlI7givdfAHx6Rli07xtkNwqaii8H/roo/mPy71454i8Mat4V1RrDW7R7eUfdY8rIPVW6EVkV3zpwqx1J2PuizvLbULSO6sZ47iCQZSSJgysPqKmr4w8M+Ndf8I3Pm6HqEkKE5eBjujf6qePx617L4a/aI0+4VIfFOnyWkmObi1+dD9V6j8M151TCTjrHUq57VRWLo3jHw74gRW0jWLS4Zv8AlmJAH+m081tVyNNOzGFFFFIAooooAKKwNa8deGfDysdV1m1idQT5Svvc/wDAVya8t8T/ALRUKK8HhPTWkboLq84X6hByfxI+lawo1J7ILns+o6nZaRYyXmqXUVpbRjLSysFA/wA+leEfED48y3iy6b4L3wQn5Xv3GHYf7A/h+p5+leVeIfFeteKrz7Trt/LdMPuoThEHoqjgVj16FLCRjrLVk3HSSPLI0krs7sSWZjkk+pNT2Gn3eqX8Nlp8D3FzOwSOKMZLE1q+FfBus+MdSFnotq0gBHmzNxHEPVm/p1r6d8AfDTSfAlmGiAutTkXE1468/wC6o/hX9T3rWtXjSXmJIzfhb8LbfwRZC+1EJPrU6YdxysCn+Bf6nv8ASvRaKK8ic5TlzSLCiiioAKKKKACiiigAooooAKKKKACiiigAr5a+PP8AyVS5/wCvaH/0GvqWq81hZ3EnmT2kEr9Nzxgn8zW1Gr7KXNYTPhqivuL+ydO/58LX/vyv+FH9k6d/z4Wv/flf8K7Prq/lFY+Ha9z/AGbP+PnX/wDcg/m9e3/2Tp3/AD4Wv/flf8KlgtLa13fZreKHd18tAufyrOrilUg42HYmpsv+pf8A3TTqK4Rnwm/+sb60lfcX9lad/wA+Fr/35X/Cj+ydO/58LX/vyv8AhXpfXV/KTY+HaK+4v7J07/nwtf8Avyv+FH9k6d/z4Wv/AH5X/Cj66v5Qscr8H/8AklOi/wDXN/8A0Y1drTYoo4YxHCixovRUGAPwp1efKXNJsoKKKKkAooooA+av2h/+R/tf+vFP/QmryevuaaxtLl99xawysBjdJGGOPxqP+ydO/wCfC1/78r/hXfTxahBRtsKx8O0V9xf2Tp3/AD4Wv/flf8KP7J07/nwtf+/K/wCFX9dX8orHz3+zn/yO2o/9eB/9DWvo+oILK1tnLW1tDCxGCY4wpI/Cp6461T2k+awwooorEYV5r8ev+SWz/wDX1D/OvSqZNBFcR+XcRJKmc7XUMPyNXCXJJS7AfCtFfcX9k6d/z4Wv/flf8KP7J07/AJ8LX/vyv+Fd/wBdX8pNj4dq5o//ACHLH/r4j/8AQhX2t/ZOnf8APha/9+V/wpRpWnggiwtgRyCIV4/Sj66v5QsWqKKK80o+OfiRpx0v4ka5bFdo+1tIo/2X+Yfo1cxX3LLp9nPIZJ7SCRz1Z4gSfxIpn9k6d/z4Wv8A35X/AAr0Y4yySaJsfDtFfcX9k6d/z4Wv/flf8KP7J07/AJ8LX/vyv+FP66v5Qsedfs/XIm+GjRZ5t72VMemQrf8As1eoVHBbwWyFbaGOFSckRqFBP4VJXBUlzycu5QVm+ItDtfEnh680i+XMN1EUzjlD2Ye4OD+FaVFSm07oD4f1rSbnQtau9Lvk2z2spjcY647j2PWqVfcsun2c8hkntIJHPVniBJ/Eimf2Tp3/AD4Wv/flf8K9BY3TWJNj4dor7i/snTv+fC1/78r/AIUf2Tp3/Pha/wDflf8ACn9dX8oWPnf4GePf7D1v/hHtTmxYag/7hmPEUx4H4N0+uPevpOqo0rT1YEWNsCDkEQrx+lWq461SNSXMlYaCiiisRhRRRQAV8U+Lf+R01v8A7CE//oxq+1qqtpdg7FnsbZmY5JMKkk/lXRQreybdriaufDlFfcX9k6d/z4Wv/flf8KP7J07/AJ8LX/vyv+FdX11fyisfDtfUHwB/5Jiv/X5L/wCy16F/ZOnf8+Fr/wB+V/wqeGCK3j2W8SRJnO1FCj8hWNbEqrHlsNIkr5v/AGi/+R4sP+vBf/Q2r6QqCeytblw1zbQysBgGSMMQPxrGjU9nPmsB8M0V9xf2Tp3/AD4Wv/flf8KP7J07/nwtf+/K/wCFdn11fyisfDtFfcX9k6d/z4Wv/flf8KP7J07/AJ8LX/vyv+FH11fyhY+HaK+4v7J07/nwtf8Avyv+FH9k6d/z4Wv/AH5X/Cj66v5QsfDtFfcX9k6d/wA+Fr/35X/Cj+ydO/58LX/vyv8AhR9dX8oWPh2ve/2bP+PXXv8Afh/k1ez/ANk6d/z4Wv8A35X/AAqaC1t7XP2aCKHd18tAufyrKrilUg42HYloooriGFFFFABRRRQBm674e0vxLpr2Gt2cd1A3QOOUPqp6g+4rwbxn8AdS05pLvwlKdQtuT9lkIEyD0B6N+h+tfRdFbU606fwisfDF5ZXWn3T219by286HDRyoVYfgagr7Y13wtoniW38nXNNguwPus6/Ov0Ycj8DXlev/ALOmn3BaTw5qstoe0N0vmL9NwwR+tehDGQfxaCsfPqsysCpII6EGt3TPHPijSNq6fr1/Ci9E89mUf8BORXRav8EvGulljHp6X8YPD2kobP8AwE4P6VyN74c1rTWIv9Jvbfb1MkDKB+OK6VKE+qYj7M0G4luvDum3Fw5eWW1id3I+8xQEn86v1meGuPCmk54/0KH/ANAFadeC9yzB8cX9zpngTWb6wlMNzb2ckkUigEqwHB5r5K1Lxl4k1gMupa5f3CN1Rp22/wDfIOK+rviMC3w18QBQSTYy4AH+zXyZY+Fte1Nwtho99cFuhS3Yj88V6ODUeVtksyiSxyTk+9JXoukfA3xpqhUz2cOnRt1a6lAI/wCArk/pXo3h/wDZ30ezZZfEOozag46xQr5SfnyT+ldMsRTj1FY+f9O0y+1e8S00y0mu7h/uxwoWJ/KvY/Bf7P1zcGO88ZTfZ4uosYGy7f7zdF+gyfpXt+jeH9J8PWv2fRdPt7KPuIkwW+p6n8a0a4qmMlLSGhVilpWj6foenx2OkWkVpbRj5Y4lwPqfU+5q7RRXDe+4wooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACkKhhhgCPQilooAAMDA4FFFFAAQGBBGQeoNAAAwBgUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAf//Z)

Software Engineering Department  
Braude College  
Capstone Project Phase A – 61998

**Rethinking Graph Neural Networks for Anomaly Detection**

24-1-R-13  
  
**Project Members**

Maayan Sharvit 318301744

[Maayan.sharvit@e.braude.ac.il](mailto:Maayan.sharvit@e.braude.ac.il)

Shir Cohen 207581471

[Shir.cohen@e.braude.ac.il](mailto:Shir.cohen@e.braude.ac.il)

**Project Supervisors**

Dr. Renata Avros

**Table Of Contents**

[**1.** **Introduction** 3](#_Toc165819516)

[**2.** **Background And Related Work** 4](#_Toc165819517)

[**2.1.** **Graph Data** 4](#_Toc165819518)

[**2.2.** **Supervised And Unsupervised Training** 4](#_Toc165819519)

[**2.3.** **Multi-Layer Perceptron** 5](#_Toc165819520)

[**2.4.** **Laplacian** 6](#_Toc165819521)

[**2.5.** **Fourier Transform** 6](#_Toc165819522)

[**2.6.** **Hammond’s Graph Wavelet Theory** 7](#_Toc165819523)

[**2.7.** **Beta Wavelet Function on Graph** 7](#_Toc165819524)

[**2.9.** **Band Pass Filter** 8](#_Toc165819525)

[**2.10.** **Graph Neural Networks** 9](#_Toc165819526)

[**3.** **Expected Achievements** 9](#_Toc165819527)

[**4.** **The Model** 10](#_Toc165819528)

[**4.1.** **The Flow of The Project** 11](#_Toc165819529)

[**4.2.** **Training The Beta Wavelet Graph Neural Network and Data Division** 13](#_Toc165819530)

[**4.3.** **Product diagrams and GUI** 14](#_Toc165819531)

[**5.** **Verification Plan** 18](#_Toc165819532)

[**6.** **Conclusion** 19](#_Toc165819533)

[**7.** **References** 19](#_Toc165819534)

**Abstract**Graph Neural Networks (GNNs) have emerged as a powerful paradigm for anomaly detection in graph-structured data. A critical design choice in GNNs is the selection of an appropriate spectral filter, which significantly influences the model's expressive power. This work presents a novel approach that analyzes anomalies through the lens of the graph spectrum. Key observation is that the presence of anomalies in the graph data manifests as a spectral shift phenomenon. This phenomenon is characterized by a rightward shift in the distribution of spectral energy, indicating a reduced emphasis on low frequencies and a heightened presence of high frequencies within the graph spectrum. Motivated by this observation, the Beta Wavelet Graph Neural Network (BWGNN) is introduced. BWGNN incorporates spectral and spatial localized band-pass filters, specifically designed to effectively capture the right-shift phenomenon associated with anomalous graph structures. The efficacy of BWGNN is validated on four large-scale anomaly detection datasets.

**Keywords:** Graph Neural Networks, Wavelet transform, Multi-layer perceptron, the Beta kernel function, band-pass filters, spatial domain, spectral domain, graph Laplacian, Fourier transform.

# **Introduction**

Nowadays data analysis, the identification of anomalies plays a pivotal role across various domains such as cyber security, fraud detection, health monitoring, and device failure detection. An anomaly is defined as a data object that significantly deviates from the majority by being abnormal within a dataset. As the commonness of graph data escalates in the Web era, the utilization of graph data has become essential in detecting instances of fraud, particularly evident in social networks and financial platforms where relationships and transactions are crucial indicators.

Graph Neural Networks (GNNs) have emerged as efficient tools for mining data, offering a natural framework for graph anomaly detection. However, traditional GNNs create challenges, especially the over-smoothing issue, wherein the aggregation of information from node neighborhoods results in the blending of anomaly representations with benign data. Therefore, distinguishing anomalies becomes difficult, leading to suboptimal performance of vanilla GNNs.

To address these challenges, various GNN models have been proposed, categorized into three groups: those employing attention mechanisms, utilizing resampling strategies, and adding auxiliary losses. All these methods analyze the anomaly detection from the graph spatial domain, but despite advancements in the spatial domain analysis, research in this domain remains relatively weak. On the other hand, there are few works that address this problem from the spectral domain. Nevertheless, choosing a tailored spectral filter is a key component of GNN design.

A substantial amount of effort was expended in exploring the question: "How can we choose a tailored spectral filter in GNNs for anomaly detection?" Taking the first step towards analyzing anomalies through the lens of the graph spectrum, we observe a phenomenon termed the 'right-shift' which is when the spectral energy distribution concentrates less on more on high frequencies with increasing anomaly degree. We validate this phenomenon and justify the necessity of spectral localized band-pass filters in graph anomaly detection.  
Building upon these insights, we introduce the Beta Wavelet Graph Neural Network (BWGNN) to address the 'right-shift' phenomenon of graph anomalies. Unlike existing works with adaptive filters, BWGNN leverages Hammond’s graph wavelet theory to propose the Beta kernel, offering multiple flexible, spatial/spectral-localized, and band-pass filters tailored specifically for anomaly detection.

Furthermore, to enable future research in this direction, two large-scale real-world datasets are introduced: the T-Finance dataset based on transaction networks and the T-Social dataset based on social networks. Accompanied by these datasets, comprehensive experiments are conducted on four datasets, employing both supervised and semi-supervised methodologies. BWGNN demonstrates superior performance over conventional graph neural networks and fine anomaly detection methods. This contributes to advancing anomaly detection techniques but also provides valuable resources for the research community.

# **Background And Related Work**

## **Graph Data**

Refers to data that is represented in the form of a graph, which consists of a collection of nodes and edges between these nodes. Graphs are a mathematical abstraction used to represent relationships or connections between entities.

In the context of anomaly detection, graph data could refer to various types of networks or relational data where nodes represent entities (such as users, transactions, or objects) and edges represent relationships or interactions between these entities. Examples of graph data include social networks (where nodes represent users and edges represent friendships), citation networks (where nodes represent academic papers and edges represent citations between papers), and many others.

Graph data is often analyzed using graph-based algorithms and techniques to extract meaningful insights, detect patterns, or identify anomalies within the network structure.

## **Supervised And Unsupervised Training**

Supervised and unsupervised training are two fundamental approaches in machine learning for training models to perform tasks and make predictions.

Supervised learning, the training data consists of input-output pairs, where the model learns to map input data to corresponding output labels.

During the training, the model is provided with input data along with the correct labels. The model adjusts its parameters to minimize the difference between its predictions and the true labels.

Unsupervised learning, the training data consists of input data without corresponding output labels. The model learns to find patterns, structures, or relationships in the data without explicit guidance. During training, the model identifies basic structures or clusters in the input data based on similarities or differences between data points.

## **Multi-Layer Perceptron**

A Multi-Layer Perceptron () is a type of artificial neural network that consists of multiple layers of interconnected nodes (neurons). It is a feedforward neural network, meaning that information flows in one direction, from the input layer through one or more hidden layers to the output layer. s are widely used for various tasks in machine learning and pattern recognition.Layer Structure:

* Input Layer: This layer contains neurons representing the input features of the data. Each neuron corresponds to a feature, and the values fed into these neurons serve as the inputs to the network.
* Hidden Layers: These are intermediate layers between the input and output layers. Each hidden layer consists of multiple neurons that perform computations on the input data.
* Output Layer: The output layer produces the final predictions or outputs of the network.

Activation Function: Each neuron in an applies an activation function to its weighted inputs to produce an output. Common activation functions include the sigmoid function and the ReLU function.

Weighted Connections: Connections between neurons in adjacent layers are associated with weights that represent the strength of the connection.

Training: s are typically trained using supervised learning algorithms.

Input Layer

Output Layer

Hidden Layers

Fig. 1: An example of a Multi-Layer Perceptron.

## **Laplacian**

The Laplacian determines a [spectral layout](https://en.wikipedia.org/wiki/Spectral_layout) of a [graph](https://en.wikipedia.org/wiki/Graph_drawing).

where is the degree matrix and is the adjacency matrix.  
The diagonal will have the degree of each node and for the rest:

if the value is 1 then the nodes are connected and 0 if the nodes are not.

* In the context of graph theory, the Laplacian matrix is a square matrix that represents a graph. It provides information about the graph's structure and connectivity.
* For a graph with N nodes, the Laplacian matrix is an N times N symmetric matrix.

The Laplacian operator plays a crucial role in spectral graph analysis. The Laplacian matrix of a graph is often used to characterize the graph's structure and properties. In the context of spectral graph analysis, the Laplacian matrix is employed to perform operations such as graph signal processing and graph convolution.

Specifically, the Laplacian matrix is utilized to compute the graph Fourier transform, which transforms graph signals from the spatial domain to the spectral domain. This transformation allows the analysis of graph data in terms of its frequency components, analogous to the Fourier transform in signal processing.

Moreover, the Laplacian matrix also governs the behavior of graph wavelet functions, which are used to extract localized information from the graph in both the spatial and spectral domains. By operating on the eigenvalues and eigenvectors of the Laplacian matrix, graph wavelets can capture important structural features and patterns of the graph data at different scales.

## **Fourier Transform**

The Fourier transform decomposes a function (a signal) into its constituent frequencies. This is particularly useful in analyzing signals in the frequency domain.   
In signal processing, the Fourier transform is used for tasks such as filtering, demodulation, and spectral analysis. It allows us to understand the frequency content of a signal.

The signal x on the graph goes through a Fourier transform to get its spectral representation. This is computed as:

,

where hat is the graph Fourier transform of and is the matrix of eigenvectors of the Laplacian matrix and.  
In other words, ​ tells us how much of the signal's energy is aligned with the frequency represented by the eigenvector ​.  
In our case, BWGNN uses the Fourier transform in a wavelet mode.

## **Hammond’s Graph Wavelet Theory**

The graph wavelet transform, as proposed by Hammond, utilizes a set of wavelets based on a "mother" wavelet ψ. These wavelets, denoted as  when applied to a graph signal , the application of can be expressed as:

which utilizes the Fourier transform,  
Here, represents a kernel function defined in the spectral domain on the interval   
, where is the maximum eigenvalue of the graph Laplacian and the function is a diagonal matrix with the values of along its diagonal and represents a diagonal matrix where each element along the diagonal corresponds to an eigenvalue (λ) of the Laplacian matrix () for the graph.

While the equation above resembles the graph spectral convolution derived from the Fourier transform, Hammond's graph wavelet transform imposes additional requirements on the kernel function :

* The wavelet transform must satisfy the condition, which is derived from the Parseval theorem:

,

This condition implies that , indicating that the wavelet behaves like a band-pass filter in the spectral domain.

* The wavelet transform needs to cover different frequency bands through band-pass filters of varying scales .

To avoid the need for eigen-decomposition of the graph Laplacian , the kernel function is often chosen to be a polynomial function, commonly.

## **Beta Wavelet Function on Graph**

The Beta wavelet function on a graph, also known as the Beta kernel function, is a type of wavelet function used in graph signal processing. It is derived from the Beta distribution and  
 is applied in the spectral domain to analyze signals defined on graphs.

The probability density function of the Beta distribution.

,

where .

As the eigenvalues of the normalized graph Laplacian satisfy we adopt to cover the complete spectral range of .

In order to achieve fast computation to ensure is a polynomial. Thus, the Beta wavelet transform can be written as

.

Let be a constant and the Beta wavelet transform is constructed by a group of Beta wavelets with the same order: .

In the equation, is a low-pass filter and others are band-pass filters of different scales.

* 1. **Cross Entropy Loss Function**

The cross-entropy loss function is used when adjusting model weights during training. It compares each predicted class probability to the actual class desired and calculates a score/loss that penalizes the probability based on how far it is from the actual expected value. The penalty is logarithmic in nature, yielding a score close to 1 for large differences and a score close to 0 for small differences.  
A perfect model has a cross-entropy loss of zero. The cross-entropy loss increases as the predicted probability deviates from the actual label:

.

## **Band Pass Filter**

A band-pass filter is a type of signal processing filter that allows signals within a specific frequency range, or "band", to pass through while blocking frequencies outside this range. In other words, it selectively passes signals with frequencies lying within the desired band and rejects signals with frequencies outside this band.

Band-pass filters are widely used in various applications, including audio processing and radio frequency communication. They are essential in separating signals of interest from unwanted noise or interfering signals, enabling efficient signal transmission, reception, and processing in systems.

תמונה שמכילה תרשים, טקסט, קו, מלבן

התיאור נוצר באופן אוטומטי

Fig. 2: An example of different frequency pass filters.

## **Graph Neural Networks**

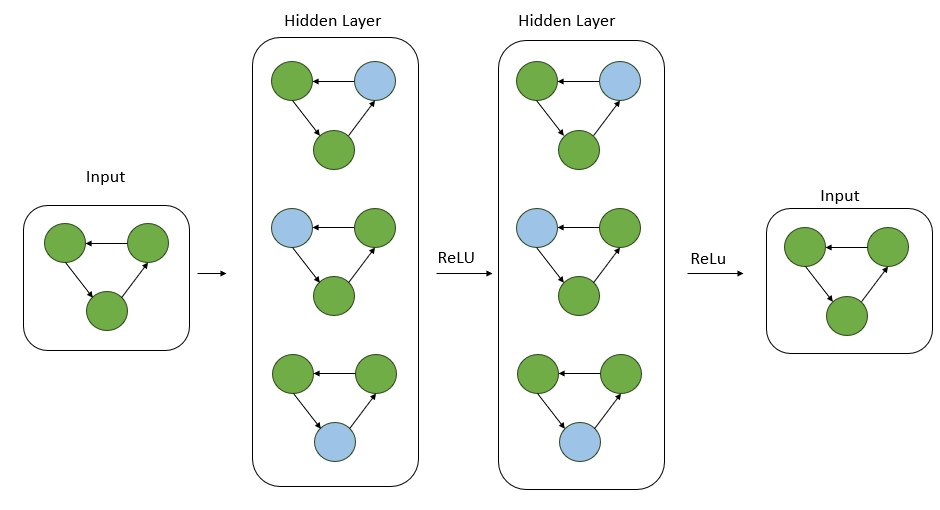
Graph Neural Networks (GNNs) are a specialized type of neural network architecture designed for processing and learning from graph-structured data, where entities are represented as nodes and relationships between them as edges.  
GNNs operate by iteratively aggregating information from neighboring nodes, allowing them to capture both local and global graph structure. They excel in tasks such as node classification, link prediction, and graph classification by leveraging this learned representation. GNNs operate in a message-passing framework, where nodes update their representations based on information passed from neighboring nodes. Their adaptability to varying graph structures and ability to capture complex patterns make them highly flexible and applicable across domains like social networks and recommendation systems.   
Overall, GNNs represent a powerful approach for analyzing and understanding relational data, contributing significantly to the field of machine learning and artificial intelligence.  


Fig. 3: An example for Graph Neural Networks.

# **Expected Achievements**

There is an anticipation of enhanced anomaly detection performance, particularly in scenarios with high-frequency anomalies, owing to the utilization of the Beta distribution to generate band-pass filters. This methodological innovation is expected to lead to more precise identification and localization of anomalies within complex graphs. Furthermore, an application of the extension of graph wavelet theory is expected to have wide-ranging implications across various domains, including finance and social networks, where anomaly detection is of paramount importance.

# **The Model**

The Beta Wavelet Graph Neural Network (BWGNN) is an architecture for anomaly detection in graph-structured data. Its core innovation lies in the utilization of the Beta wavelets, meticulously designed spectral filters. These filters target the "right-shift" phenomenon, a hallmark of anomalies within the graph spectrum. Traditional GNNs often struggle with anomaly detection due to their reliance on low-pass or adaptive filters. These filters might not adequately capture the unique spectral signatures of anomalies. BWGNN addresses this limitation by employing the Beta wavelets in parallel for each signal. These wavelets, band-pass and spectral-localized properties, excel at capturing high-frequency anomalies that might evade traditional methods.  
The BWGNN architecture draws inspiration from Hammond's graph wavelet theory, enabling the construction of diverse band-pass filters with varying scales. This empowers BWGNN to focus on the specific spectral features indicative of anomalous behavior, leading to superior anomaly detection capabilities.   
By effectively harnessing the power of the Beta wavelets and parallel filtering, BWGNN offers a robust and efficient framework for identifying anomalies within graph data.

תמונה שמכילה תרשים, ציור, שרטוט, קו

התיאור נוצר באופן אוטומטי

Fig. 4: The process of the model.

## **The Flow of The Project**

The BWGNN analyzes the graph data through parallel filtering with the Beta wavelets, capturing different frequency bands potentially associated with anomalies. It then combines the filtered information and calculates a probability score for each node, reflecting the likelihood of an anomaly. By minimizing the loss function during training, the system learns to distinguish normal nodes from anomalies within the graph structure.

* A crucial step involves labeling the data. This labeling process essentially creates a training ground for the model to learn the difference between normal and anomalous behavior within the graph.
* The input for BWGNN consists of information about the nodes in the graph we are analyzing for anomalies.
* The output of the first serves to transform the input graph features into a higher-level representation that captures important information about the graph's structure and attributes. Essentially, acting as a feature extractor, the extracts meaningful features from the raw input data, which are then passed on to subsequent layers for further processing and analysis. Utilizing the non-linear transformations provided by the 's hidden layers and activation functions, the model can effectively capture complex relationships and patterns within the graph data, facilitating tasks such as anomaly detection.
* After the first , wavelet functions are activated to enhance the model's ability to capture localized and multi-scale information within the graph. These wavelet functions serve as filters that allow the model to focus on specific regions of the graph while analyzing it at various scales simultaneously. By decomposing the graph signal into different frequency components using wavelets, the model can extract relevant spectral features that are indicative of anomalies.
* After the wavelet functions are applied, the resulting features are aggregated. There are two aggregation functions mentioned in the paper: summation or concatenation. The results from the wavelet transform are aggregated and represented as .
* The anomaly probability is calculated by using a with a sigmoid activation function applied on . The purpose of this second is to further process and refine the extracted features to make a final decision regarding the presence of anomalies in the graph. The sigmoid activation function is commonly used in binary classification tasks, such as anomaly detection, where the model needs to output a probability score indicating the likelihood of each data point being anomalous. By applying the sigmoid activation function, the model can produce output probabilities that are bounded between 0 and 1, with higher values indicating a higher likelihood of anomaly. Overall, the second serves as a classifier that utilizes the extracted features to make informed decisions about the presence or absence of anomalies in the graph.
* Through this iterative process, the BWGNN refines its skills and becomes adept at identifying anomalies in unseen graph data.
* After training, use the BWGNN model to analyze new unseen graph data.

The model assigns an anomaly score to each node, allowing for the identification of potential anomalies within the graph.

## **Training The Beta Wavelet Graph Neural Network and Data Division**

All models except SVM undergo 100 epochs of training utilizing the Adam optimizer with a learning rate set at 0.01, with the best Macro-F1 model during validation being preserved. For the YelpChi, Amazon, and T-Finance datasets, the dimensionality 'h' for both representations and hidden states is uniformly configured to 64 across all models, while the parameter 'C' in BWGNN is designated as 2. Within BWGNN, concatenation serves the function of .   
Data division entails a 40% training ratio for supervised learning and 1% for semi-supervised learning, with the remaining portion divided into a 1:2 ratio for validation and testing. In the T-Social dataset, 'h' is similarly set to 64, 'C' to 5, with the supervised training ratio at 40% and a semi-supervised ratio of 0.01% (incorporating solely 17 labeled anomalies). Validation and test sets maintain a 1:2 ratio. Results are presented as the average value and standard deviation of 10 runs for YelpChi and Amazon, whereas for T-Finance and T-Social, averages are derived from 5 runs with distinct random seeds to ensure that the initialization of model parameters and the shuffling of data points are varied across runs.   
Moreover, it's essential to highlight that the data comes unlabeled, necessitating manual labeling for training purposes.

supervised learning

Semi- supervised learning

Data

1:2

validation and testing

1%

40%

59%

## **Product diagrams and GUI**

The following Sequence Diagram represents the BWGNN’s workflow:

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי

First, the User activates the BWGNN and provides unseen graph data containing node features. The BWGNN then iterates through each node: it processes features using an MLP, applies the Beta wavelets for frequency analysis, aggregates the wavelet results, feeds them into a final MLP, and finally calculates an anomaly score between 0 and 1. After processing all nodes, the BWGNN returns the anomaly scores back to the User.   
This sequence diagram highlights the core steps involved in the BWGNN model's analysis of unseen graph data to identify potential anomalies.

תמונה שמכילה טקסט, צילום מסך, גופן, תצוגה

התיאור נוצר באופן אוטומטי

Fig. 5: Home window, training a new model task is selected.

תמונה שמכילה טקסט, צילום מסך, גופן, תצוגה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן, תצוגה

התיאור נוצר באופן אוטומטי

Fig. 6: Model Training.

Fig. 7: Home window – Detect an anomaly task is selected.

תמונה שמכילה טקסט, צילום מסך, תצוגה, גופן

התיאור נוצר באופן אוטומטי

Fig. 8: Detect an anomaly by uploading a graph.

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

Fig. 9: Anomaly table as an output from the BWGNN.

תמונה שמכילה צילום מסך, טקסט, תרשים, עיצוב

התיאור נוצר באופן אוטומטי

Fig. 10: Visual result from BWGNN, highlights the nodes that are labeled as anomalies.

# **Verification Plan**

Predicted scenarios to ensure the system meets specified requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| Case | Name | Description | Expected Output |
| 1 | Wrong input | the input is not node features. | An error message appears on the screen. |
| 2 | Activating anomaly detection without pre-training | The user starts using the model for anomaly detection before the model is trained. | An error message appears on the screen. |
| 3 | Wrong aggregation result | The C beta wavelets results aggregated poorly and gave a wrong result. | An error message appears on the screen. |
| 4 | Invalid value of epoch's amount variable | The input for the epoch's variable is negative. | An error message appears on the screen. |
| 5 | Missing output | The last fails and the probability prediction is not shown. | An error message appears on the screen. |
| 6 | Trying to predict too many times. | The user activates anomaly detection too many times with the same dataset. | An error message appears on the screen. |

# **Conclusion**

The conclusion of the study sheds light on a novel understanding of anomalies within graphs, particularly focusing on their behavior in the spectral domain. Through their analysis, the researchers identified a distinct phenomenon termed the 'right-shift', which describes how anomalies influence the spectral energy distribution of graphs. This observation provides valuable insights into the spectral characteristics of anomalies, highlighting potential patterns that can be leveraged for detection. In response to this phenomenon, the researchers introduced a new approach called the Beta Wavelet Graph Neural Network (BWGNN). This model utilizes the Beta graph wavelets to generate band-pass filters with strong locality properties, aiming to effectively capture anomaly information within graphs. Empirical validation using diverse datasets demonstrated that BWGNN outperformed existing methods in terms of both accuracy and scalability, showcasing its potential as a robust solution for anomaly detection in graph data. Overall, the conclusion emphasizes the significance of their findings in advancing anomaly detection techniques and introduces BWGNN as a promising tool for addressing this challenge effectively.

# **References**

[1] BWGNN; [Tang et al., 2022] Jianheng Tang, Jiajin Li, Ziqi Gao and Jia Li.  
Rethinking Graph Neural Networks for Anomaly Detection. Proceedings of the International Conference on Machine Learning (ICML), 123-135. arXiv:2205.15508

[2] Wavelets on Graphs via Spectral Graph Theory; [Hammond et al., 2009] David K Hammond\*, a,1, Pierre Vandergheynstb,2, and R´emi Gribonval.  
Applications of the transform through examples of wavelets on graphs corresponding to a variety of different problem domains.  
arXiv:0912.3848v1 [math.FA] 19 Dec 2009

[3] MLOps blog; [Menzli, 2023] Aman Menzli.  
Graph Neural Network and some GNN applications: Everything you need to know.  
https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications

[4] Medium; [Lalf, 2023] Lalf.  
From Graph Fourier Transform to Graph Convolution.  
https://medium.com/@lalf\_klein/from-graph-fourier-transform-to-graph-convolution-b32c691b92fd

[5] MLOps blog; [Czakon, 2023] Jakub Czakon.  
F1 score vs. ROC AUC vs. accuracy vs. PR AUC: which evaluation metric should you choose?  
<https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc>

[6] DataCamp; [Jaiswal, 2024] Sejal Jaiswal.  
Multilayer Perceptrons in Machine Learning: A Comprehensive Guide. https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning